UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

PROYECTO FIN DE CARRERA

Clasificación estructural de proteínas

Autor: Álvaro Téllez Rodríguez
Tutor: Valentín Moreno Pelayo

Octubre 2015

A mi familia. En especial, a mi padre.

"Sólo cabe progresar cuando se piensa en grande. Sólo es posible avanzar cuando se mira lejos." (José Ortega y Gasset)

Agradecimientos

Hace ya más de 9 años desde que aprobé el último examen de la carrera. En estos años, además de en los anteriores, ha habido mucha gente que directa o indirectamente ha colaborado en la realización de este trabajo, bien por su ayuda o consejo, bien por las lecciones que con sus actos me han enriquecido a nivel personal. De todos ellos quisiera acordarme de:

Los profesores que tuve durante toda la carrera. Tanto a los buenos como a los menos buenos. Los buenos me enseñaron muchas cosas, me despertaron muchos intereses que no sabía que tenía, y me mostraron una pasión por el conocimiento envidiable. De los menos buenos aprendí que siempre hay esforzarse para llegar a alcanzar lo que se desea, y que no siempre tendrá premio el esfuerzo, pero siempre es un premio el propio esfuerzo invertido, lección que me empezó a preparar para el mundo laboral.

Mi tutor, Valentín, por su inmensa paciencia conmigo a lo largo de estos 9 años. Nunca ha perdido la esperanza en que este momento llegaría, y seguramente sin él no hubiera llegado.

Las camareras de la cafetería del Sabatini, principalmente –aunque no solamente–, Paqui, Esther y Carmen, que siempre fueron (y han seguido siendo las veces que he pisado por allí) simplemente geniales.

Mis compañeros de trabajo Pepe, Hans, Sonia y Jose, por su insistencia a lo largo de los últimos años en que realizase este trabajo. Sin duda sus insistencias me han servido de combustible en algunos momentos en los que justo eso era lo que necesitaba. De tantos otros compañeros de trabajo (Paco, Inma, Raquel, Juanjito, Joliva, Carmelo, Rubén, Agus…) que en estos casi 9 años me habéis aportado tanto: conocimientos técnicos, formas diferentes de afrontar los problemas, apoyo, y un largo etcétera.

Mis amigos Andrés, Luis, Narci, Raúl y Manu, por estar ahí y haberme aconsejado sabiamente cuando os he necesitado. Y a Saúl, que además de lo anterior, me ha aconsejado mucho y bien sobre la realización de este trabajo.

Y en especial de mi familia. De mi madre Mª Carmen y mi hermano Raúl, por haber sido tan *pesados* estos años, insistiendo en que debía hacer este trabajo sí o sí, que no tenía otras opciones. De mis abuelos Miguel y Tarsila (allá donde estés), por todo lo que me habéis dado siempre, y por esos macarrones con chorizo que la mayoría de semanas me disteis durante tantos años. Y en especial de mi padre, José Ángel, al que más ilusión le habría hecho verme terminar la carrera.

A todos ellos, **GRACIAS**.

Resumen

Las proteínas están compuestas por aminoácidos, que aparecen como secuencias lineales. Éstas se pliegan sobre sí mismas y con ello se forman estructuras tridimensionales. Conociendo la estructura de las proteínas se puede determinar la función de las mismas, pudiendo ser útil para el estudio de enfermedades genéticas.

Algunas veces no es sencillo determinar la estructura de una proteína y por ello algunas proteínas se han clasificado con cierta estructura aunque pudiera tener otra.

Con este trabajo se pretende buscar proteínas, que inicialmente están catalogadas como proteínas sin estructura espiral, que realmente tienen estructura espiral. Para ello se hace uso de la herramienta de aprendizaje automático Weka, para que analice los datos conocidos que se tienen de partida sobre la estructura de algunas proteínas, y a partir de esa información generar unas reglas y árboles de decisión que posteriormente se aplicarán sobre las proteínas de las que no se tiene certeza de su estructura para ver si hay alguna proteína, de las que inicialmente se supone que no tienen estructura espiral, que destaque por tener características similares a aquellas que tienen estructura espiral.

Abstract

Proteins are compoused by amino acids, which appear as a linear sequence. These amino acids collapse over theirselves creating three-dimensional structures. Knowing the protein's structure it's possible to determine the protein's application, what could be usefull for the genetic diseases exploration.

Sometimes it's not easy to determine the structure of a protein, and because of that some proteins have been classified with an structure when in fact they have another one.

With this Project we pretend to find proteins that initially are classified as they don't have spiral structure, but in fact they do. For doing this, we'll use the automatic learning tool Weka, in order to let it analyze the known data we have about some well known proteins, and with that knowledge generate rules and decisión trees that later on will be applied over proteins from which we don't know for sure their structures, trying to find some proteins that could have spiral structure.

Tabla de contenidos

Índice de figuras

Índice de tablas

1. Introducción

1.1.Objetivo

El objetivo de este trabajo es analizar algunas proteínas catalogadas como proteínas sin estructura física en forma de espiral (por sencillez, diremos que se trata de una clasificación *negativa*) para ver si alguna de ellas es un falso negativo, y por tanto que tenga estructura física en forma de espiral (por sencillez, clasificación *positiva*). Para ello se va a hacer uso de la herramienta para minería de datos Weka [19].

Se han usado los resultados del análisis de las estructuras de algunas proteínas que se saben que tienen dicha estructura en forma de espiral para que la herramienta Weka pueda formar unos modelos de reglas sobre las diferencias sub-secuencias de aminoácidos que forman las proteínas. Posteriormente se han aplicado dichos modelos sobre las proteínas inicialmente catalogadas con estructura no espiral para intentar localizar algunas proteínas que pudieran tener estructura en forma de espiral.

1.2.Alcance

Se ha partido de 129 proteínas con estructura primaria espiral y 18.603 clasificadas a priori con estructura no espiral.

Estudios previos han puesto de manifiesto que hay proteínas catalogadas como carentes de estructura espiral mal clasificadas.

Este trabajo está orientado a descubrir estos falsos negativos mediante modelos basados en aprendizaje automático por inducción de reglas. El hecho de saber la existencia de errores en los ejemplos negativos de los test de aprendizaje hace centrar el foco de la investigación en su utilidad para detectar proteínas con catalogación errónea, y no tanto en obtener modelos con altos porcentajes de acierto en los test de validación.

1.3.Estructura y contenido del documento

Este documento se divide en los siguientes capítulos:

- **Introducción**: breve aproximación al objetivo del proyecto, además de la estructura de este documento.
- **Estado del arte**: se presenta una introducción a la biología de las proteínas y aminoácidos, haciendo hincapié en la parte más relacionada con este proyecto, además de comentar algunas técnicas de aprendizaje automático y algunos trabajos previos.

- **Metodología**: estrategia seguida para la resolución del proyecto.
- **Experimentación**: se detalla el proceso seguido con los datos de las proteínas con las que se partía.
- **Resultados obtenidos**: donde se exponen los resultados obtenidos en este proceso.
- **Conclusiones y trabajos futuros**: lecciones aprendidas y propuestas para continuar este proyecto.
- **Apéndices**: bibliografía, presupuesto…

2. Estado del arte

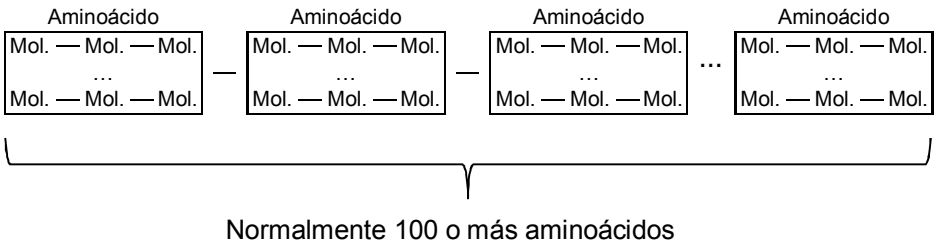
En este capítulo se va a hacer una breve introducción al mundo biológico relacionado con este proyecto, para poder entender mejor el trabajo realizado, además de una introducción a los métodos de aprendizaje automático y un resumen de algunos trabajos relacionados previos a este.

2.1.Proteínas y aminoácidos

Las proteínas son moléculas necesarias para la vida, ya que forman parte de las células de la materia vegetal y animal. Desempeñan diferentes funciones: estructural, inmunológica, enzimática, etc. [21], [27]

Los aminoácidos son moléculas orgánicas que se combinan entre sí formando péptidos (2 aminoácidos forman un dipéptido, 3 aminoácidos forman un tripéptido, etc). Cuando el péptido está formado por un número elevado (en general, más de 10) de aminoácidos se llama polipéptido. Y cuando el polipéptido es suficientemente grande (normalmente al menos 100 aminoácidos) se le llama proteína [22], [24], [25]. De forma más visual, el esquema de una proteína:

Ilustración 1: Esquema de la estructura de una proteína



No todos los aminoácidos que existen pueden componer una proteína. Los aminoácidos que componen las proteínas que pueden aparecer en el código genético son:

Tabla 1: Listado de aminoácidos que pueden aparecer en las proteínas del código genético [18]

Nombre	Abreviatura	
Ácido aspártico	Asp	D
Ácido glutámico	Glu	E
Alanina	Ala	A
Arginina	Arg	R
Asparagina	Asn	N
Cisteína	Cys	C

Nombre	Abreviatura	
Fenilalanina	Phe	F
Glicina	Gly	G
Glutamina	Gln	Q
Histidina	His	H
Isoleucina	Ile	I
Leucina	Leu	L
Lisina	Lys	K
Metionina	Met	M
Prolina	Pro	P
Serina	Ser	S
Tirosina	Tyr	Y
Treonina	Thr	T
Triptófano	Trp	W
Valina	Val	V

2.2.Estructura de las proteínas

La estructura de las proteínas se divide en cuatro niveles:

- **Estructura primaria**: se refiere a la secuencia de aminoácidos que forma la proteína. Este nivel de estructura influye directamente en la función de la proteína y en la forma que esta adopte.
- **Estructura secundaria**: es la que define la forma en 3 dimensiones de la proteína (forma espiral, etc). Tiene lugar cuando los aminoácidos de la estructura primaria interactúan entre sí.
- **Estructura terciaria**: forma geométrica de los átomos que forman la proteína. Se define como una secuencia de una o más estructuras secundarias distintas.
- **Estructura cuaternaria**: cuando la proteína está formada por más de una cadena de aminoácidos

Es importante conocer la estructura de las proteínas ya que de ella se puede deducir el comportamiento de la misma, lo que ayuda, entre otras cosas, en la investigación de enfermedades raras [23], [26], [28]. Y no siempre es sencillo determinar la estructura de las proteínas.

2.3.Trabajos previos asociados

Como trabajos previos asociados destacan 2:

Por un lado el artículo Functional and Genomic Analyses of Alpha-Solenoid Proteins [1] en el que trata de encontrar estructuras que contengan alfa-solenoides en proteínas en la que inicialmente no se habían detectado, haciendo uso de redes neuronales para ello. De este

trabajo surge la idea de intentar localizar proteínas que tenga cierta estructura sobre un conjunto de proteínas que inicialmente no la tenía.

Por otro lado, la tesis *Análisis de los criterios de relevancia documental mediante consultas de información en el entorno web* [14] que usa la herramienta de aprendizaje automático Weka para, partiendo de cierta información, intentar determinar qué campos son más importantes de cara al posicionamiento Web en motores de búsqueda. De este trabajo viene la idea de usar la herramienta Weka y así hacer uso del aprendizaje automático para intentar encontrar esas posibles proteínas que pudieran tener una estructura que inicialmente no se había detectado en ellas.

2.4.Punto de partida

Este proyecto empieza a raíz del artículo Functional and Genomic Analyses of Alpha-Solenoid Proteins [1], con la idea de usar algoritmos de inducción de reglas en lugar de redes neuronales, para intentar encontrar proteínas con estructura espiral en un conjunto de proteínas que inicialmente están catalogadas como que no tienen estructura espiral. Se parte de uno de los ficheros de salida expuesto en el artículo que contiene proteínas con estructura espiral.

La idea es intentar obtener algunas proteínas que inicialmente se han catalogado, sin haberse estudiado en profundidad, como proteínas sin estructura espiral y que en realidad sí pudieran tener estructura espiral.

2.5.Técnicas de aprendizaje automático

Las técnicas de aprendizaje automático permiten recopilar datos analizando un conjunto de datos de prueba (o entrenamiento) para que a partir de ellos pueda localizar patrones y generar reglas que den lugar a esos patrones, con la idea de poder aplicarlos posteriormente a otros datos y así poder determinar más información de la que anteriormente se tenía [14].

Usaremos en este trabajo los siguientes algoritmos de aprendizaje:

- **PART**: Este clasificador usa la estrategia separar-y-vencer (separate-and-conquer). Construye un árbol de decisión en cada iteración y hace de la mejor hoja una regla [2].

- **J48**: Genera un árbol de decisión C4.5 podado (o no podado) [3].

- **NBTree**: genera un árbol de decisión con clasificadores Bayes en las hojas [4].

- **AdaBoostM1**: algoritmo de aprendizaje automático adaptativo que en su generación va corrigiendo valores mal clasificados anteriormente [5], [6] y [7]. Se puede combinar con otros algoritmos para mejorar los resultados. En este trabajo se combinará con los algoritmos PART, J48 y NBTree.

- **Bagging**: de bootstrap aggregating. Método de generación de múltiples versiones de un predictor, usando un predictor agregado [8]. Este algoritmo se puede combinar con otros algoritmos para mejorar los resultados. En este trabajo se combinará con los algoritmos PART, J48 y NBTree.

- **RacedIncrementalLogitBoost**: clasificador para ficheros de gran tamaño. Para ello divide la entrada en trozos y por cada uno de ellos genera un clasificador [9].

- **RandomCommittee**: conjunto de clasificadores. Cada uno se genera usando una semilla de números aleatorios. El modelo final es un promedio de las predicciones generadas [11].

- **RotationForest**: formado por varios árboles de decisión, cada uno de ellos formado a partir de un subconjunto aleatorio de la características de entrada [12] y [13]. Este algoritmo se debe combinar con otros algoritmos. En este trabajo se combinará con los algoritmos PART y J48.

Se toman estos algoritmos por parecer buenos clasificadores según lo visto en otros trabajos [14] o que destacan en las comparaciones de clasificadores [15], [16], [17], además de por la forma de clasificar de algunos algoritmos, que parecen buenos dadas las condiciones de este trabajo.

A priori creemos que los mejores algoritmos para este trabajo serán:

- **RacedIncrementalLogitBoost**, por tratarse de un clasificador preparado para ficheros de gran tamaño, que al dividir el fichero de entrada y generar diferentes clasificadores parece más robusto ante posibles malas clasificaciones puntuales.

- **RandomCommittee**, por ser un promedio de varias predicciones, ya que parece que las posibles clasificaciones erróneas no contarán debido a que deberían aparecer pocas veces en relación a las clasificaciones correctas.

- **RotationForest**, por tratarse de varios árboles de decisión, lo cual hará que las clasificaciones correctas hayan sido confirmadas varias veces.

Adicionalmente a la generación de los modelos también vamos a hacer uso de algunos evaluadores de atributos para determinar qué atributos o posiciones de aminoácidos parecen más importantes de cara a determinar la estructura de la sub-secuencia de la proteína. Estos evaluadores serán:

- **ChiSquaredAttributeEval**: evalúa la relación entre el valor de clase y cada atributo por medio del valor estadístico chi cuadrado [29].

- **GainRatioAttributeEval**: evalúa la importancia de los atributos midiendo el ratio de beneficio respecto a la clase [30].

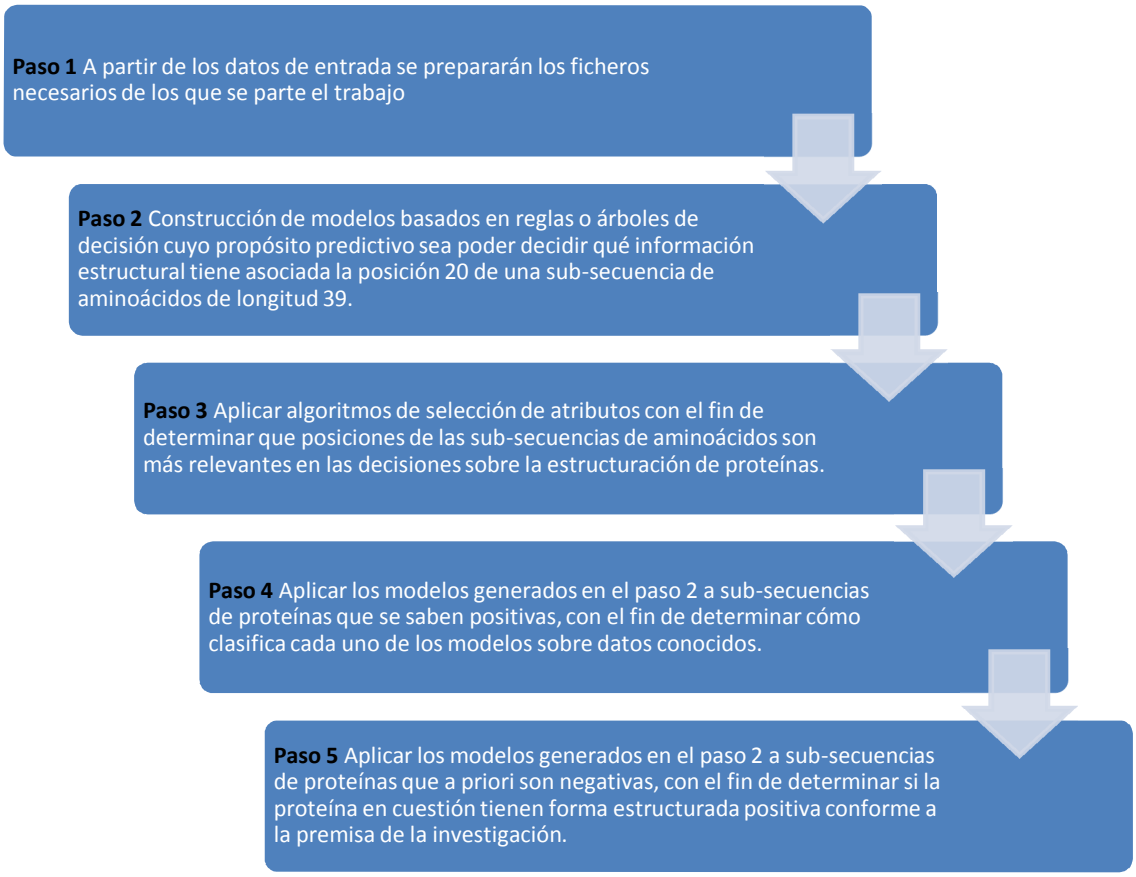
- **InfoGainAttributeEval**: evalúa la importancia de los atributos midiendo el beneficio que ofrece respecto a la clase [31].

- **OneRAttributeEval**: evalúa la importancia de los atributos usando el clasificador OneR [32].

3. Metodología

A grandes rasgos, los pasos que se van a seguir para la realización de este proyecto son, y en este orden:

Figura 1: Principales pasos a seguir para la realización de este trabajo



Detallando un poco más cada uno de los pasos:

3.1.Preparación de ficheros a partir de los datos de entrada

Datos de entrada	Fichero generado con esos datos
Fichero con proteínas con y sin forma estructurada, cuya estructura final es conocida, y por cada una de las sub-secuencias que componen a las proteínas se conoce si dicha sub-secuencia confiere forma espiral o no	Fichero en formato apto para la herramienta Weka. Este fichero será el fichero de entrenamiento a partir del cual se podrán generar los modelos posteriormente

Fichero con proteínas que se saben positivas (tienen estructura espiral), pero de las que no se conoce cuál o cuáles de las sub-secuencias confiere dicha estructura espiral	Fichero en formato apto para la herramienta Weka. Este fichero se usará para poner a prueba cada uno de los modelos generados
Fichero con proteínas que inicialmente están catalogadas como negativas (sin estructura espiral) pero de las que no se ha realizado un análisis completo. No se conoce información estructural sobre cada una de las sub-secuencias	Fichero en formato apto para la herramienta Weka. Sobre este fichero se aplicarán cada uno de los modelos generados con la intención de buscar proteínas candidatas a ser positivas (tener estructura espiral)

3.2.Construcción de modelos de decisión

A partir del fichero de entrenamiento generado en formato Weka se generarán diferentes modelos de decisión.

Los algoritmos usados son:

1. Part
2. J48
3. NBTree
4. AdaBoostM1. Este algoritmo además se combina con estos otros algoritmos:
    a. Part
    b. J48
    c. NBTree
5. Bagging. Este algoritmo además se combina con estos otros algoritmos:
    a. Part
    b. J48
    c. NBTree
6. RacedIncrementalLogitBoost
7. RandomCommittee
8. RotationForest. Este algoritmo sólo se puede utilizar en combinación de otro algoritmo. En este trabajo se ha combinado con estos otros algoritmos:
    a. Part
    b. J48

En total se van a generar 15 modelos.

3.3.Selección de atributos más determinantes a la hora de determinar la estructura de cada sub-secuencia

Con la intención de determinar qué posiciones, dentro de las sub-secuencias de aminoácidos, son las más importantes se van a aplicar algunos algoritmos sobre el fichero de entrenamiento.

3.4.Aplicar los modelos sobre el fichero de proteínas positivas

Se van a aplicar los modelos generados en el paso 2 en el fichero de proteínas que sabemos que son positivas (tienen estructura espiral), aplicando las reglas/árbol de decisión de dichos modelos a todas las sub-secuencias de 39 aminoácidos de todas las proteínas.

Posteriormente veremos cuántas sub-secuencias de cada una de las proteínas están clasificadas como positivas y así poder analizar finalmente cómo trabaja cada uno de los modelos (detectan muchos o pocos positivos en total, por cada proteínas, etc.).

3.5.Aplicar los modelos sobre el fichero de proteínas negativas

Se van a aplicar los modelos generados en el paso 2 en el fichero de proteínas inicialmente catalogadas como negativas (sin estructura espiral), aplicando las reglas/árbol de decisión de dichos modelos a todas las sub-secuencias de 39 aminoácidos de todas las proteínas.

Posteriormente veremos cuántas sub-secuencias de cada una de las proteínas están clasificadas como positivas para así buscar proteínas candidatas a ser positivas.

4. Experimentación

4.1.Datos de los que se parte

A continuación se detallan cada uno de los ficheros tomados de entrada, indicando su procedencia y el proceso seguido para obtener el fichero final usado:

4.1.1. Fichero de entrenamiento para Weka, base para la generación de modelos

Partimos de uno de los ficheros publicados en el artículo functional and Genomic Analyses of Alpha-Solenoid Proteins[1] que contiene la definición de 28 proteínas que se usarán como conjunto de entrenamiento del proceso. Estas 28 proteínas tienen estructura conocida por los investigadores de dicho artículo.

Este fichero está completado con la secuencia de aminoácidos de cada una de las proteínas, y con la información de estructura asociada a cada parte de la secuencia. Esta información ha sido proporcionada por el Grupo de investigación Computational Biology and Data Mining dirigido por el doctor Miguel Andrade[2].

Transformamos este fichero en otro fichero de formato válido para Weka, que contiene sub-secuencias de 39 aminoácidos (cada aminoácido se conoce, en este fichero Weka, como atributo) de las 28 proteínas que había de partida. Cada secuencia de aminoácidos lleva asociada información estructural {0,1} (siendo: 0=estructura no espiral, 1=estructura espiral). Para esta transformación se ha tenido en cuenta una premisa en la que se basa esta investigación, que consiste en: "Si existe una subsecuencia de aminoácidos de longitud 39 en una proteína y en dicha subsecuencia su posición 20 tiene asociada información estructural con valor 1 se concluye que la proteína tiene una forma estructurada", es decir, dividimos la secuencia de aminoácidos que compone la proteína en sub-cadenas de 39 aminoácidos, y se comprueba si en el aminoácido situado en el medio de la sub-cadena, el aminoácido en la posición 20, tiene asociado valor de estructura espiral en el fichero de entrenamiento de proteínas con estructuras conocidas.

[1] Fichero con listado de proteínas disponible en internet, bajo este enlace: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0079894#pone.0079894.s004.

[2] Grupo situado en el centro MDC MAX-DELBRÜCK-CENTER FOR MOLECULAR MEDICINE BERLIN-BUCH. Más información en la página web http://cbdm.mdc-berlin.de/

Una pequeña muestra de este fichero resultante:

Figura 2: muestra del fichero de entrenamiento en formato válido para Weka

```
E,V,M,V,K,V,A,S,R,H,A,N,E,L,P,A,I,L,E,D,L,N,L,S,V,P,Q,R,A,A,H,L,N,A,L,K,M,N,C,0
V,M,V,K,V,A,S,R,H,A,N,E,L,P,A,I,L,E,D,L,N,L,S,V,P,Q,R,A,A,H,L,N,A,L,K,M,N,C,F,0
M,V,K,V,A,S,R,H,A,N,E,L,P,A,I,L,E,D,L,N,L,S,V,P,Q,R,A,A,H,L,N,A,L,K,M,N,C,F,I,1
V,K,V,A,S,R,H,A,N,E,L,P,A,I,L,E,D,L,N,L,S,V,P,Q,R,A,A,H,L,N,A,L,K,M,N,C,F,I,L,0
K,V,A,S,R,H,A,N,E,L,P,A,I,L,E,D,L,N,L,S,V,P,Q,R,A,A,H,L,N,A,L,K,M,N,C,F,I,L,T,0
```

Este fichero tiene 34.696 sub-secuencias de aminoácidos consideradas negativas (valor estructural 0), y tan sólo 260 consideradas positivas (valor estructural 1).

Para que no esté tan descompensado el número de secuencias negativas y positivas se han repetido 133 veces las sub-secuencias positivas, y así se ha conseguido balancear la muestra.

Al final nos queda otro fichero con 34.696 sub-secuencias de aminoácidos consideradas negativas y 34.580 sub-secuencias de aminoácidos consideradas positivas. Este fichero será el que se tomará como entrada para nuestro proceso como base para generar los modelos.

4.1.2. Fichero de proteínas positivas

Tomamos como conjunto de proteínas positivas las mismas que en el artículo functional and Genomic Analyses of Alpha-Solenoid Proteins[3]. Este fichero contiene únicamente el identificador[4] de la proteína y su descripción:

Tabla 2: Ejemplo de información contenida en el fichero de proteínas positivas del artículo functional and Genomic Analyses of Alpha-Solenoid Proteins

PDB ID	Information
1B3U_A	CRYSTAL STRUCTURE OF CONSTANT REGULATORY DOMAIN OF HUMAN 2 PP2A, PR65ALPHA

Este fichero lo completamos con la secuencia de aminoácidos de cada una de las proteínas, que obtenemos de uno de los bancos de datos de proteínas (PDB, por sus siglas en inglés)[5], y así generamos nuestro fichero de proteínas positivas en formato FASTA, teniendo 2 líneas por cada proteína (una para el nombre de la proteína, y la otra para la secuencia de aminoácidos que la compone). Una muestra de este fichero:

[3] Fichero disponible en internet, bajo el enlace http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0079894#pone.0079894.s001

[4] Llamado PDB ID, que significa Protein Data Bank Identification, que es un código internacional establecido para identificar, entre otras, a las proteínas. Más información en http://www.wwpdb.org/ o en alguno de los bancos de información establecidos, como el europeo http://www.ebi.ac.uk/pdbe/.

[5] Por ejemplo, para la proteína 1B3U podemos ir a la página http://www.ebi.ac.uk/pdbe/entry/pdb/1b3u, y desde ahí navegar a la página http://www.ebi.ac.uk/pdbe/entry/pdb/1b3u/protein/1, donde tenemos la secuencia de aminoácidos de la proteína en formato FASTA.

Figura 3: muestra del fichero en formato FASTA de proteínas positivas

```
>1B3U_A
AAADGDDSLYPIAVLIDELRNEDVQLRLNSIKKLSTIALALGVERTRSELLPFLTDTIYDEDEVLLALAEQLGTFTTLVGGPEYVHCLLPPLES
LATVEETVVRDKAVESLRAISHEHSPSDLEAHFVPLVKRLAGGDWFTSRTSACGLFSVCYPRVSSAVKAELRQYFRNLCSDDTPMVRRAAASKL
GEFAKVLELDNVKSEIIPMFSNLASDEQDSVRLLAVEACVNIAQLLPQEDLEALVMPTLRQAAEDKSWRVRYMVADKFTELQKAVGPEITKTDL
VPAFQNLMKDCEAEVRAAASHKVKEFCENLSADCRENVIMSQILPCIKELVSDANQHVKSALASVIMGLSPILGKDNTIEHLLPLFLAQLKDEC
PEVRLNIISNLDCVNEVIGIRQLSQSLLPAIVELAEDAKWRVRLAIIEYMPLLAGQLGVEFFDEKLNSLCMAWLVDHVYAIREAATSNLKKLVE
KFGKEWAHATIIPKVLAMSGDPNYLHRMTTLFCINVLSEVCGQDITTKHMLPTVLRMAGDPVANVRFNVAKSLQKIGPILDNSTLQSEVKPILE
KLTQDQDVDVKYFAQEALTVLSLA
>1DL2_A
GAGEMRDRIESMFLESWRDYSKHGWGYDVYGPIEHTSHNMPRGNQPLGWIIVDSVDTLMLMYNSSTLYKSEFEAEIQRSEHWINDVLDFDIDAE
VNVFETTIRMLGGLLSAYHLSDVLEVGNKTVYLNKAIDLGDRLALAFLSTQTGIPYSSINLHSGQAVKNHADGGASSTAEFTTLQMEFKYLAYL
TGNRTYWELVERVYEPLYKNNDLLNTYDGLVPIYTFPDTGKFGASTIRFGSRGDSFYEYLLKQYLLTHETLYYDLYRKSMEGMKKHLLAQSKPS
SLWYIGEREQGLHGQLSPKMDHLVCFMGGLLASGSTEGLSIHEARRRPFFSKSDWDLAKGITDTCYQMYKQSSSGLAPEIVVFNDGNIKQDGWW
RSSVGDFFVKPLDRHNLQRPETVESIMFMYHLSHDHKYREWGAEIATSFFENTCVDCNDPKLRRFTSLSDCITLPTKKSNNMESFWLAETLKYL
YILFLDEFDLTKVVFNTEAHPFPVLDEEILKSQSLTTGWSL
>1DVP_A
MFRSSFCKNLENATSHLRLEPDWPSILLICDEINQKDVTPKNAFAAIKKKMNSPNPHSSCYSLLVLESIVKNCGAPVHEEVFTKENCEMFSSFL
ESTPHENVRQKMLELVQTWAYAFRSSDKYQAIKDTMTILKAKGHTFPELREADAMFTADTAPNWADGRVCHRCRVEFTFTNRKHHCRNCGQVFC
GQCTAKQCPLPKYGIEKEVRVCDGCFAALQRG
```

4.1.3. Fichero de proteínas negativas

Como conjunto inicial de proteínas negativas (sin estructura espiral) partimos de un fichero con formato FASTA que contiene 19.647 proteínas inicialmente catalogadas como negativas, proporcionado por el Grupo de investigación Computational Biology and Data Mining dirigido por el doctor Miguel Andrade[6].

En este fichero hay varias proteínas que contienen en su secuencia el valor X, que se usa de comodín para referir que puede ser cualquier elemento (ver el punto del anexo valores válidos para un aminoácido en el formato FASTA). Para que los modelos funcionen mejor, eliminamos estos elementos en las secuencias en las que aparezcan.

Una vez eliminados esos elementos, también debemos eliminar del fichero aquellas proteínas que tengan menos de 39 de aminoácidos, ya que el proceso seguido toma la secuencia de 39 en 39 aminoácidos.

Al final nos queda un fichero de 18.603 proteínas en formato FASTA inicialmente catalogadas como negativas y válidas para el proceso. Éste será nuestro conjunto de proteínas negativas. Una pequeña muestra de este fichero:

```
>12as:A
MKTAYIAKQRQISFVKSHFSRQLEERLGLIEVQAPILSRVGDGTQDNLSGAEKAVQVKVKALPDAQFEVVHSLAKWKRQTLGQ…
>16vp:A
SRMPSPPMPVPPAALFNRLLDDLGFSAGPALCTMLDTWNEDLFSALPTNADLYRECKFLSTLPSDVVEWGDAYVPERTQIDIR…
>1914:A
MASMTGGQQMGRIPGNSPRMVLLESEQFLTELTRLFQKCRSSGSVFITLKKYDGRTKPIPRKSSVEGLEPAENKCLLRATDGK…
```

[6] Grupo situado en el centro MDC MAX-DELBRÜCK-CENTER FOR MOLECULAR MEDICINE BERLIN-BUCH. Más información en la página web http://cbdm.mdc-berlin.de/

4.2.Objetivos buscados con estos datos

A partir de los datos anteriores, se pretende:

1) Decidir en qué casos una sub-secuencia de aminoácidos confiere a una proteína una forma estructurada.
2) Determinar que posiciones de las sub-secuencias son más relevantes en las decisiones.
3) Localizar candidatos a proteínas con forma estructurada de entre las que se consideran no estructurales.

4.3.Preparación de datos

4.3.1. Ficheros de proteínas positivas y negativas para Weka

4.3.1.1. Generación de fichero de proteínas positivas

Se parte del fichero de entrada en formato FASTA de proteínas positivas mencionado en el punto Fichero de proteínas positivas

Este fichero lo transformamos en un fichero con formato válido para Weka con el proceso descrito en el punto Proceso para transformar un fichero FASTA en un fichero para Weka. Al final nos queda un fichero con formato ARFF[7] válido para Weka.

El fichero contiene todas las sub-secuencias de 39 aminoácidos de cada proteína y el indicador de clase (indicador usado para determinar la estructura de la proteína), que como es desconocido se le pone el valor '?', usado en Weka como no definido.

Una muestra de este fichero resultante:

Figura 4: muestra del fichero de sub-secuencias de proteínas positivas en formato válido para Weka

```
T,Q,T,G,I,P,Y,S,S,I,N,L,H,S,G,Q,A,V,K,N,H,A,D,G,G,A,S,S,T,A,E,F,T,T,L,Q,M,E,F,?
Q,T,G,I,P,Y,S,S,I,N,L,H,S,G,Q,A,V,K,N,H,A,D,G,G,A,S,S,T,A,E,F,T,T,L,Q,M,E,F,K,?
T,G,I,P,Y,S,S,I,N,L,H,S,G,Q,A,V,K,N,H,A,D,G,G,A,S,S,T,A,E,F,T,T,L,Q,M,E,F,K,Y,?
G,I,P,Y,S,S,I,N,L,H,S,G,Q,A,V,K,N,H,A,D,G,G,A,S,S,T,A,E,F,T,T,L,Q,M,E,F,K,Y,L,?
I,P,Y,S,S,I,N,L,H,S,G,Q,A,V,K,N,H,A,D,G,G,A,S,S,T,A,E,F,T,T,L,Q,M,E,F,K,Y,L,A,?
```

4.3.1.2. Generación de fichero de proteínas negativas

Se parte del fichero de entrada en formato FASTA de proteínas negativas mencionado en el punto Fichero de proteínas negativas

Este fichero lo transformamos en un fichero con formato válido para Weka con el proceso descrito en el punto Proceso para transformar un fichero FASTA en un fichero para Weka. Al final nos queda un fichero con formato ARFF válido para Weka.

[7] Formato de fichero atributo-relación, ARFF por sus siglas en inglés (Attribute-Relation File Format). Más información en http://weka.wikispaces.com/ARFF

Este fichero contiene todas las sub-secuencias de 39 aminoácidos de cada proteína y el indicador de clase (indicador usado para determinar la estructura de la proteína), que como es desconocido se le pone el valor '?', usado en Weka como no definido.

Una muestra de este fichero resultante:

Figura 5: muestra del fichero de sub-secuencias de proteínas negativas en formato válido para Weka

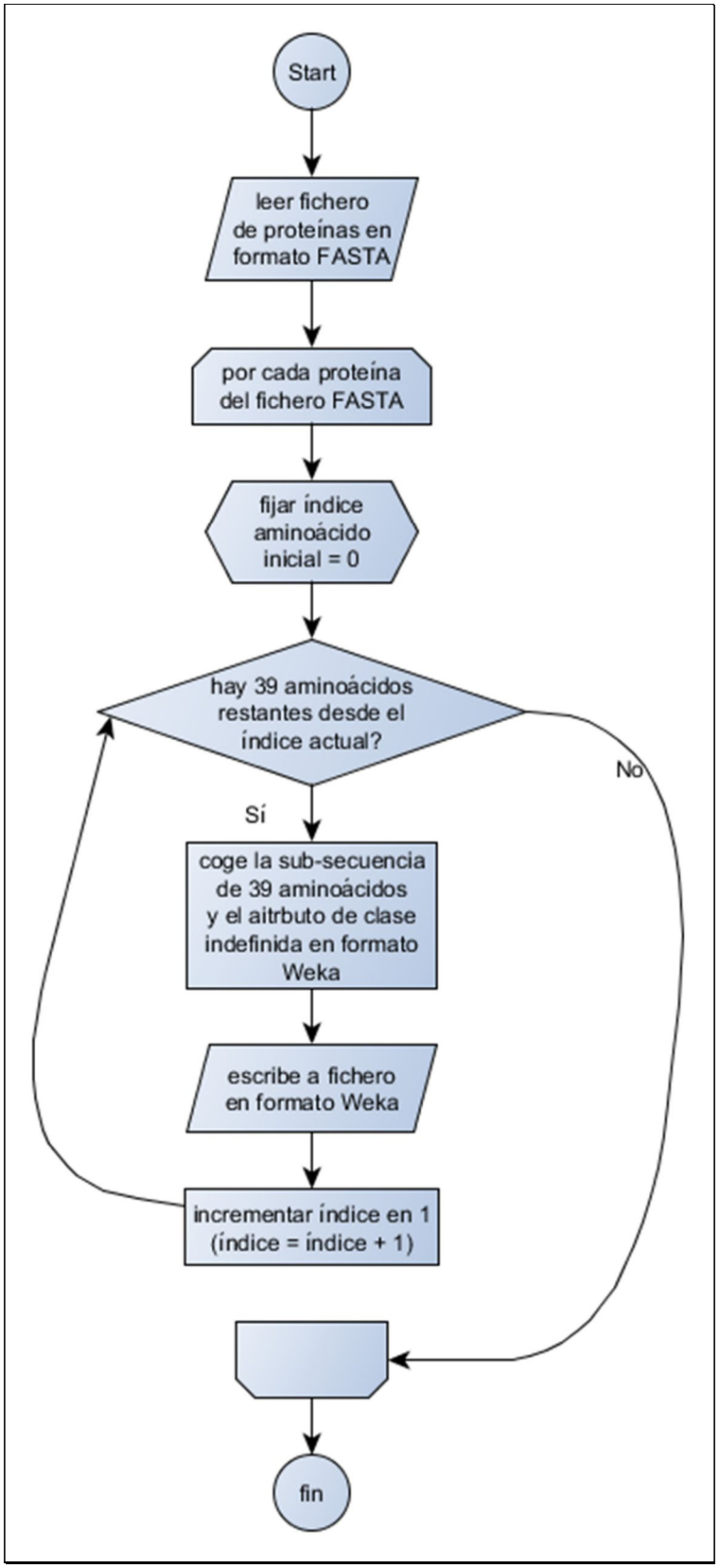
```
K,R,G,A,D,G,Y,L,L,K,D,M,E,P,E,D,L,L,K,A,L,H,Q,A,A,A,G,E,M,V,L,S,E,A,L,T,P,V,L,?
R,G,A,D,G,Y,L,L,K,D,M,E,P,E,D,L,L,K,A,L,H,Q,A,A,A,G,E,M,V,L,S,E,A,L,T,P,V,L,A,?
G,A,D,G,Y,L,L,K,D,M,E,P,E,D,L,L,K,A,L,H,Q,A,A,A,G,E,M,V,L,S,E,A,L,T,P,V,L,A,A,?
A,D,G,Y,L,L,K,D,M,E,P,E,D,L,L,K,A,L,H,Q,A,A,A,G,E,M,V,L,S,E,A,L,T,P,V,L,A,A,S,?
D,G,Y,L,L,K,D,M,E,P,E,D,L,L,K,A,L,H,Q,A,A,A,G,E,M,V,L,S,E,A,L,T,P,V,L,A,A,S,L,?
```

4.3.1.3. Proceso para transformar un fichero FASTA en un fichero para Weka

Para la transformación de un fichero FASTA a un fichero de entrada válido para Weka se ha preparado un pequeño programa. Este programa leerá la secuencia de aminoácidos que compone cada una de las proteínas y la dividirá en sub-secuencias de 39 aminoácidos (para Weka, cada uno de estos aminoácidos se conoce como atributo), añadiendo un atributo de clase como indefinido.

El diagrama UML de actividad de dicho programa:

Figura 6: Diagrama UML de actividad del programa para transformar los ficheros en formato FASTA que recibimos en un fichero de entrada para Weka



4.3.2. Modelos y reglas

A partir del fichero de entrenamiento descrito en el punto Fichero de entrenamiento para Weka, base para la generación de modelos se van a generar unos modelos y reglas que serán lo que se usen posteriormente con las proteínas positivas (para determinar cómo de restrictivos son) y con las proteínas negativas (para intentar localizar los falsos positivos).

La idea es que el fichero de entrenamiento tiene la información estructural de la proteína a nivel de aminoácido, ya que son proteínas con estructuras conocidas. Por tanto, se puede usar dicha información, dividida en sub-secuencias de 39 aminoácidos, para formar unos modelos y reglas que infieran una información de estructura espiral/no-espiral. Así, si el modelo generado es suficientemente bueno, al aplicarlo sobre un fichero del que se desconoce inicialmente la información estructural de cada sub-secuencia, se determinará qué sub-secuencias son candidatas a ser positivas (espirales).

Para la generación de los modelos se han utilizado algoritmos de aprendizaje de inducción de reglas ya que son muy expresivos al proporcionar la información en la que se basan las decisiones.

Entre las ventajas que presentan estas técnicas destacan:

- Robustez frente al ruido (debidos a errores, omisiones o insuficiencia de datos).
- Identificación de atributos irrelevantes.
- Detección de la ausencia de atributos discriminantes y de vacios de conocimiento.
- Extracción de reglas fáciles de entender y de gran expresividad.
- Posibilidad de reprocesar las reglas mediante el conocimiento de expertos, interpretando, modificando o aceptando reglas (Major y Mangano, 1995).

En concreto los algoritmos que vamos a utilizar son:

- AdaBoostM1
- Bagging
- RacedIncrementalLogitBoost
- RandomCommittee
- RotationForest

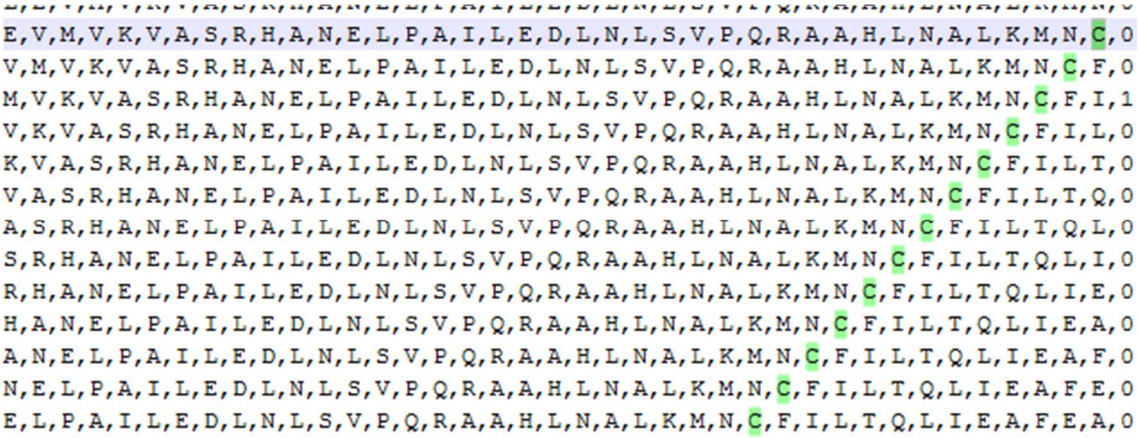
Y los vamos a combinar con los algoritmos:

- J48
- PART
- NBTree

Las instancias de aprendizaje se han obtenido tomando todas las sub-secuencias de longitud 39 de las 28 proteínas del fichero de entrenamiento descrito en Fichero de entrenamiento para Weka, base para la generación de modelos. Cada instancia contiene los 39 aminoácidos más el atributo clase {0,1} asociado al aminoácido que está en la posición 20 de la sub-secuencia.

Ejemplos de instancias:

Figura 7: Muestra de instancias del fichero de entrenamiento que se va a usar en Weka



Con el proceso descrito en el anexo Cómo generar un modelo en Weka a partir de un fichero de entrenamiento se han generado los diferentes modelos que luego se usarán en los ficheros de proteínas positivas y negativas.

Ádemás de la generación de cada modelo, con el fin de conocer las posiciones en las sub-secuencias de aminoácidos con más influencia en la estructura de las proteínas se han aplicado evaluadores de atributos individuales. Estos evaluadores al no eliminar atributos redundantes, son adecuados para proporcionar listas ordenadas de todos los atributos según su calidad, con independencia de los demás. Los evaluadores de atributos utilizados son:

- **ChiSquaredAttributeEval**: Obtiene el nivel de correlación entre la clase y cada uno de los atributo calculando el valor estadístico Chi-cuadrado.
- **GainRatioAttributeEval**: Evalúa los atributos examinando su razón de beneficio con respecto a la clase.
- **InfoGainAttributeEval**: Después de discretizar los atributos numéricos, se calcula la ganancia de información de cada atributo con respecto a la clase.
- **OneRAttributeEval**: También discretiza los atributos numéricos. Evalúa los atributos con el clasificador OneR.

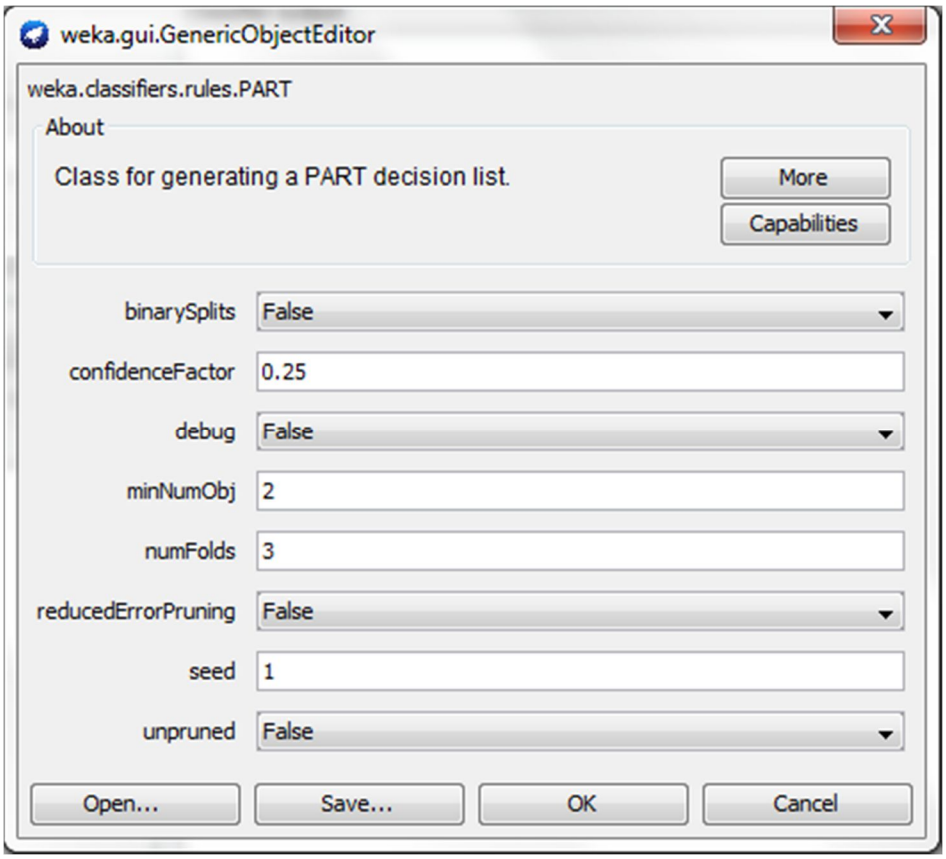
A continuación se detalla cada uno de los modelos generados y la clasificación de atributos (que se corresponden a los aminoácidos de la sub-secuencia):

Nota: todos los modelos que se detallan a continuación se han generado mediante evaluación cruzada con diez divisiones estratificadas (*10-fold cross-validation*).

4.3.2.1. Modelo Part

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 8: Configuración usada para la generación del modelo PART



Nota: el significado de cada campo se explica en el anexo Significado de los campos que se usan en la generación de un modelo en Weka.

Esta configuración se corresponda al comando:

```
weka.classifiers.rules.PART -M 2 -C 0.25 -Q 1
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero Fichero de entrenamiento para Weka, base para la generación de modelos.

Al final el modelo resultante se compone de 164 reglas. Algunas de las reglas del modelo generado:

```
P29 = K AND
P38 = S: 0 (193.0)
```

Esta regla quiere decir que si la posición 29 tiene valor *K* y la posición 38 tiene valor *S* entonces la sub-secuencia es negativa (valor 0). Esta regla se cumple, sin fallo, en 193 sub-secuencias del fichero de entrenamiento.

```
P25 = G AND
P36 = Y AND
P16 = E: 1 (133.0)
```

Si la posición 25 de la sub-secuencia tiene valor *G*, la 36 valor *Y* y la 16 valor *E* entonces la sub-secuencia es positiva (valor 1). Esta regla se cumple, sin fallo, en 133 sub-secuencias del fichero de entrenamiento.

```
P25 = G: 0 (1010.0)
```

Si la posición 25 de la sub-secuencia tiene valor *G* entonces la sub-secuencia es negativa. Esta regla se aplica posteriormente a la regla anterior. Esta regla se cumple, sin fallo, en 1.010 sub-secuencias del fichero de entrenamiento.

```
P25 = R AND
P13 = F AND
P29 = L AND
P28 = A: 1 (799.0/1.0)
```

Si la posición 25 de la sub-secuencia tiene valor *R*, la 13 valor *F*, la 29 valor *L* y la 28 valor *A*, entonces la sub-secuencia es positiva. Esta regla se cumple en 799 sub-secuencias del fichero de entrenamiento clasificando mal 1 sub-secuencia.

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 3: Matriz de confusión de la generación del modelo PART

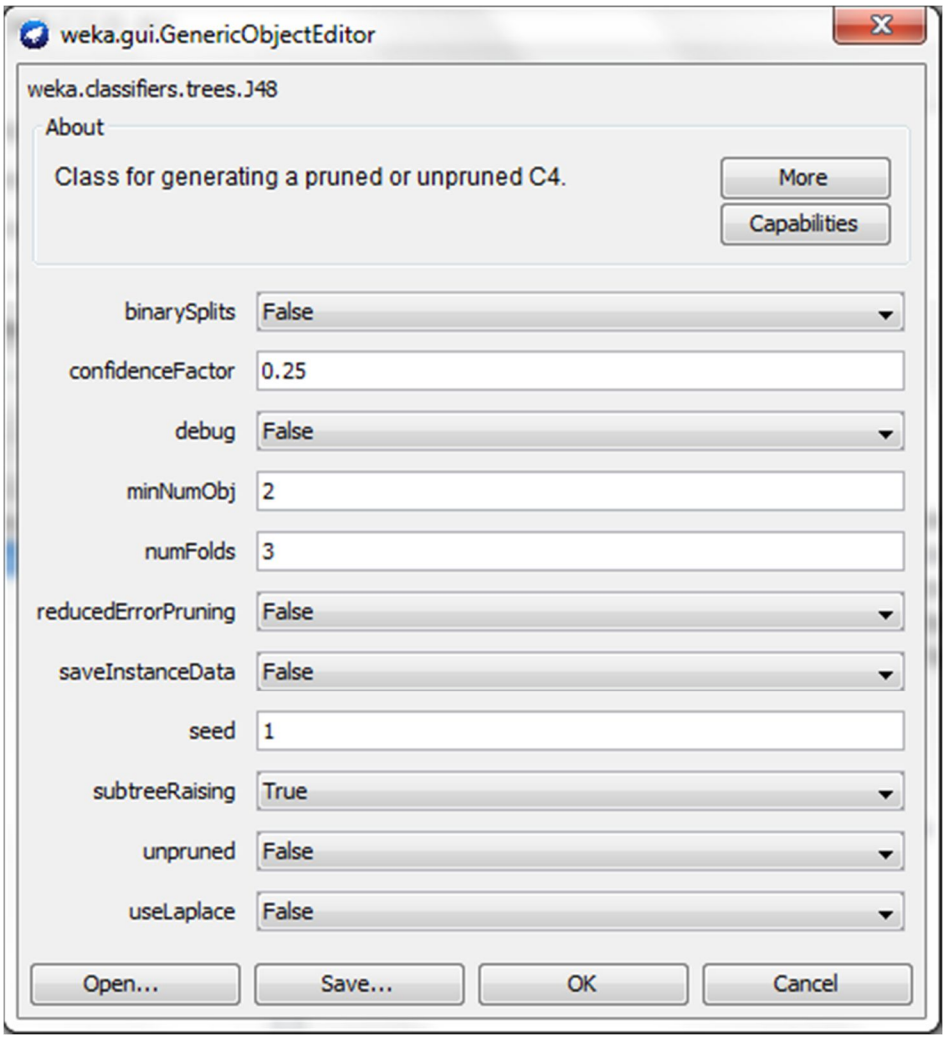
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	203 (0,585%)
	Negativas	0 (0%)	34.493 (99,415%)

Los resultados sobre las instancias positivas no son representativos ya que están repetidas, y por tanto se han utilizado ya en el proceso de aprendizaje y generación de reglas. El dato interesante de esta tabla es que este modelo ha clasificado correctamente como negativas el 99,415% (34.493 de 34.696) de las instancias que son negativas.

4.3.2.2. Modelo J48

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 9: Configuración usada para la generación del modelo J48



Nota: el significado de cada campo se explica en el anexo Significado de los campos que se usan en la generación de un modelo en Weka.

Esta configuración se corresponda al comando:

```
weka.classifiers.trees.J48 -C 0.25 -M 2
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero Fichero de entrenamiento para Weka, base para la generación de modelos.

El árbol resultante se compone de 3.174 hojas, con un tamaño total de 3.341 nodos. Una pequeña parte del árbol de decisión generado:

```
P25 = C
|   P39 = A: 0 (38.0)
|   P39 = C: 0 (14.0)
|   P39 = D: 0 (29.0)
|   P39 = E: 0 (48.0)
```

```
|   P39 = F: 0 (19.0)
|   P39 = G
|   |   P29 = A: 1 (266.0)
|   |   P29 = C: 1 (0.0)
|   |   P29 = D: 0 (1.0)
…
```

Este trozo del árbol parte de la condición que la posición 25 tenga valor *C*. A continuación evalúa el valor de la posición 39, y en función de su valor determina si la sub-secuencia es negativa (para valores *A* -38 casos en el fichero de entrenamiento-, *C* -14 casos-, *D* -29 casos-, *E* -48 casos- y *F* -19 casos-), o si debe seguir evaluando la sub-secuencia (valor *G*). En este caso evalúa la posición 29, y en función de su valor determina que la sub-secuencia es positiva (valores *A* -266 casos- y *C* -0 casos-) o negativa (valor *D* -1 caso-).

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 4: Matriz de confusión de la generación del modelo J48

		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	884 (2,548%)
	Negativas	0 (0%)	33.812 (97,452%)

Los resultados sobre las instancias positivas no son representativos ya que están repetidas, y por tanto se han utilizado ya en el proceso de aprendizaje y generación de reglas. El dato interesante de esta tabla es que este modelo ha clasificado correctamente como negativas el 97,452% (33.812 de 34.696) de las instancias que son negativas.

4.3.2.3. Modelo NBTree

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 10: Configuración usada para la generación del modelo NBTree

Este algoritmo no tiene configuraciones adicionales, por lo que el comando queda:

```
weka.classifiers.trees.NBTree
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero [Fichero de entrenamiento para Weka, base para la generación de modelos].

El árbol resultante se compone de 39 hojas (con un clasificador en cada una de ellas), con un tamaño total de 41 nodos. Algunas de las hojas:

```
P1 = A: NB 1
P1 = C: NB 2
P1 = D: NB 3
P1 = E: NB 4
P1 = F: NB 5
P1 = G
|   P4 = A: NB 7
```

Las hojas se definen en función del valor de la posición 1, y cuando ésta tiene valor *G* se evalúa la posición 4. Cada una de las hojas se compone de un clasificador compuesto por una tabla por cada uno de los posibles valores de cada una de las posiciones. Por ejemplo, la primera de las hojas (para cuando la posición 1 tiene valor *A*):

```
Leaf number: 1 Naive Bayes Classifier

                Class
Attribute           0      1
              (0.49) (0.51)
============================
P1
  A             2473.0 2528.0
  C                1.0    1.0
  D                1.0    1.0
  E                1.0    1.0
  F                1.0    1.0
  G                1.0    1.0
  H                1.0    1.0
  I                1.0    1.0
  K                1.0    1.0
  L                1.0    1.0
  M                1.0    1.0
  N                1.0    1.0
  P                1.0    1.0
  Q                1.0    1.0
  R                1.0    1.0
  S                1.0    1.0
  T                1.0    1.0
  V                1.0    1.0
  W                1.0    1.0
  Y                1.0    1.0
  [total]       2492.0 2547.0

P2
  A              252.0  267.0
  C               39.0    1.0
  D              120.0  134.0
  E              159.0  134.0
  F               84.0    1.0
  G               94.0  533.0
  H               51.0    1.0
  I              149.0  134.0
  K              188.0  134.0
  L              303.0  134.0
  M               68.0  134.0
  N               88.0  134.0
  P              104.0    1.0
  Q               99.0    1.0
  R              120.0    1.0
  S              202.0  134.0
  T              128.0  400.0
  V              182.0  134.0
  W               12.0  134.0
  Y               50.0    1.0
  [total]       2492.0 2547.0
```

```
P3
  A              213.0    1.0
  C               46.0    1.0
  D              149.0  134.0
  E              177.0  267.0
  F              106.0    1.0
  G              118.0  267.0
  H               54.0    1.0
  I              128.0  134.0
  K              171.0  267.0
  L              282.0  267.0
  M               59.0  134.0
  N               98.0    1.0
  P               93.0  267.0
  Q              108.0  400.0
  R              135.0    1.0
  S              201.0  400.0
  T              123.0    1.0
  V              160.0    1.0
  W               12.0    1.0
  Y               59.0    1.0
  [total]       2492.0 2547.0
…
```

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 5: Matriz de confusión de la generación del modelo NBTree

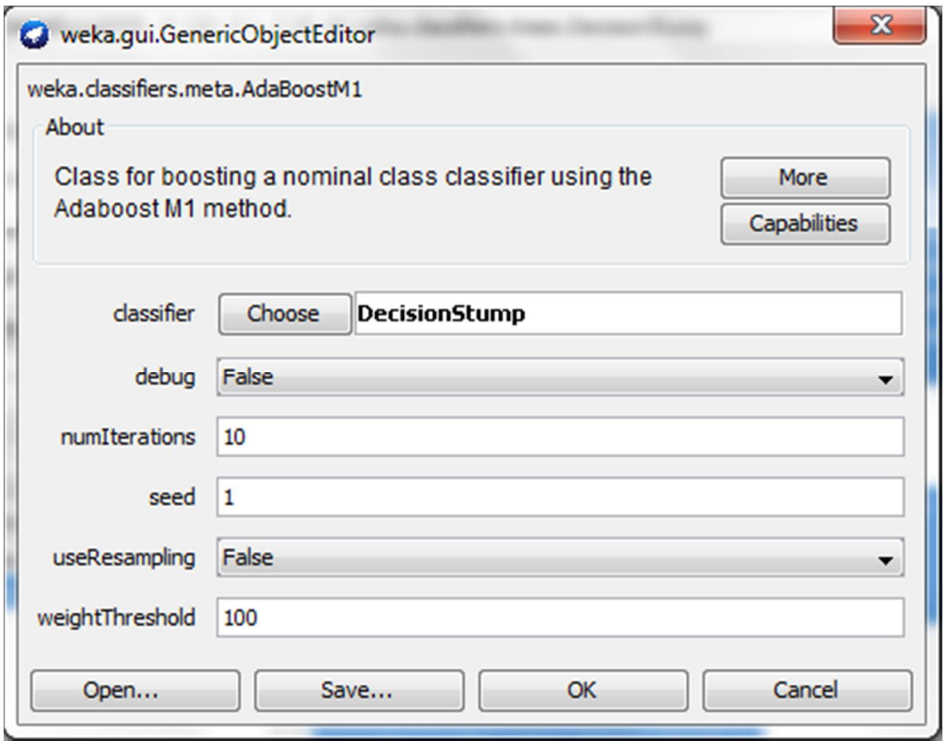
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	2 (0,006%)
	Negativas	0 (0%)	34.694 (99,994%)

Los resultados sobre las instancias positivas no son representativos ya que están repetidas, y por tanto se han utilizado ya en el proceso de aprendizaje y generación de reglas. El dato interesante de esta tabla es que este modelo ha clasificado correctamente como negativas el 99,994% (34.694 de 34.696) de las instancias que son negativas.

4.3.2.4. Modelo AdaBoostM1

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 11: Configuración usada para la generación del modelo AdaBoostM1



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.DecisionStump
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero Fichero de entrenamiento para Weka, base para la generación de modelos.

El modelo resultante se compone de 10 clasificadores con diferentes pesos. Uno de estos clasificadores:

```
Classifications

P29 = A : 1
P29 != A : 0
P29 is missing : 0

Class distributions

P29 = A
0       1
0.16817234190410008    0.8318276580959
P29 != A
0       1
0.5880552417738586     0.41194475822614146
P29 is missing
0       1
0.5008372307869969     0.49916276921300307


Weight: 0.57
```

Este clasificador evalúa la posición 29 de la sub-secuencia, y el peso que tiene esta posición cuando tiene valor *A* o no (porque en este trabajo no aplica la parte de decisión de que esa posición no tenga valor).
Cuando la posición 29 tiene el valor *A*, la probabilidad de que la sub-secuencia sea negativa es del 16,817…%, y de ser positiva del 83,182…%.
Cuando tiene un valor distinto de *A*, la probabilidad de que la sub-secuencia sea negativa es del 58,805…%, y de ser positiva del 41,194…%.
Este clasificador tiene un peso de 0,57.

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 6: Matriz de confusión de la generación del modelo AdaBoostM1

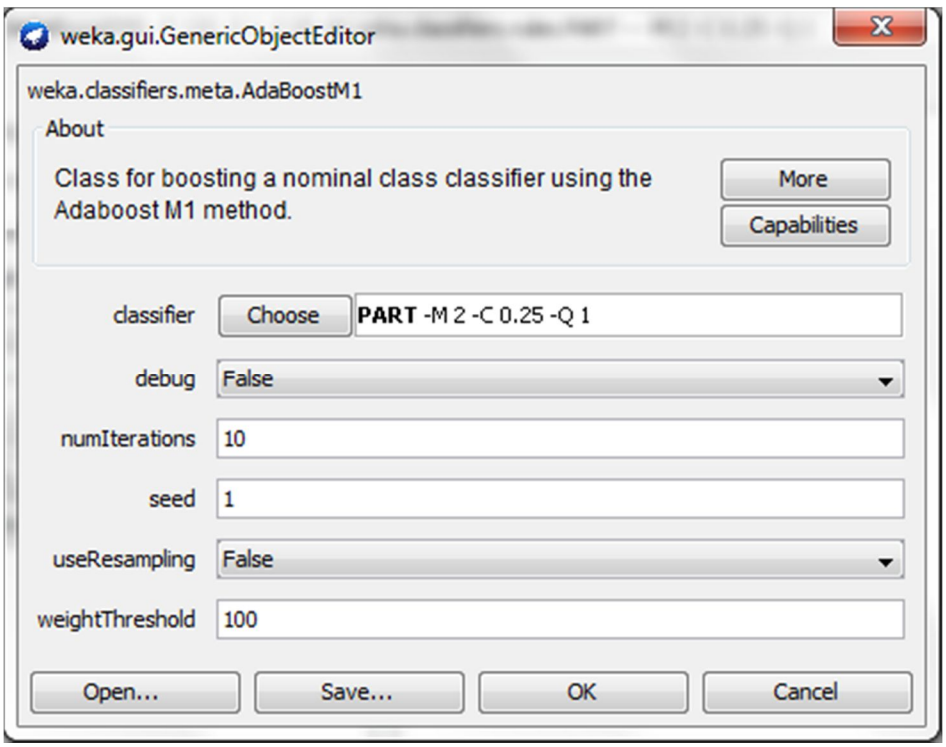
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	23.807 (68,846%)	2.644 (7,621%)
	Negativas	10.773 (31,154%)	32.052 (92,379%)

Los resultados sobre las instancias positivas nos dan una idea de que este modelo quizás no sea el que mejor evalúe ya que dichas instancias están repetidas, por lo que es más sencillo poder catalogarlas. En cualquier caso, este modelo ha clasificado correctamente como negativas el 92,379% (32.052 de 34.696) de las instancias que son negativas.

4.3.2.5. Modelo AdaBoostM1-Part

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 12: Configuración usada para la generación del modelo AdaBoostM1-Part



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C
0.25 -Q 1
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero Fichero de entrenamiento para Weka, base para la generación de modelos.

El modelo resultante tiene 162 reglas. Algunas de ellas:

```
P25 = W: 0 (234.0)
```

Si la posición 25 tiene valor *W* entonces la sub-secuencia es negativa. Esta regla se cumple, sin fallo, en 234 sub-secuencias del fichero de entrenamiento.

```
P29 = K AND
P38 = L: 0 (276.0)
```

Si la posición 29 tiene valor *K* y la posición 38 tiene valor *L* entonces la sub-secuencia es negativa. Esta regla se cumple, sin fallo, en 276 sub-secuencias del fichero de entrenamiento.

```
P25 = R AND
P13 = I AND
P24 = V AND
P10 = L: 1 (668.0/3.0)
```

Si la posición 25 tiene valor *R*, la posición 13 tiene valor *I*, la posición 24 tiene valor *V* y la posición 10 tiene el valor *L* entonces la sub-secuencia es positiva. Esta regla se cumple en 668 sub-secuencias del fichero de entrenamiento, y en 3 clasifica de forma incorrecta (es decir, que en 3 casos la sub-secuencia es negativa).

```
P13 = K AND
P20 = H AND
P17 = A: 1 (133.0)
```

Si la posición 13 tiene valor *K*, la posición 20 tiene valor *H* y la posición 17 tiene valor *A* entonces la sub-secuencia es positiva. Esta regla se cumple, sin fallo, en 133 sub-secuencias del fichero de entrenamiento.

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 7: Matriz de confusión de la generación del modelo AdaBoostM1-Part

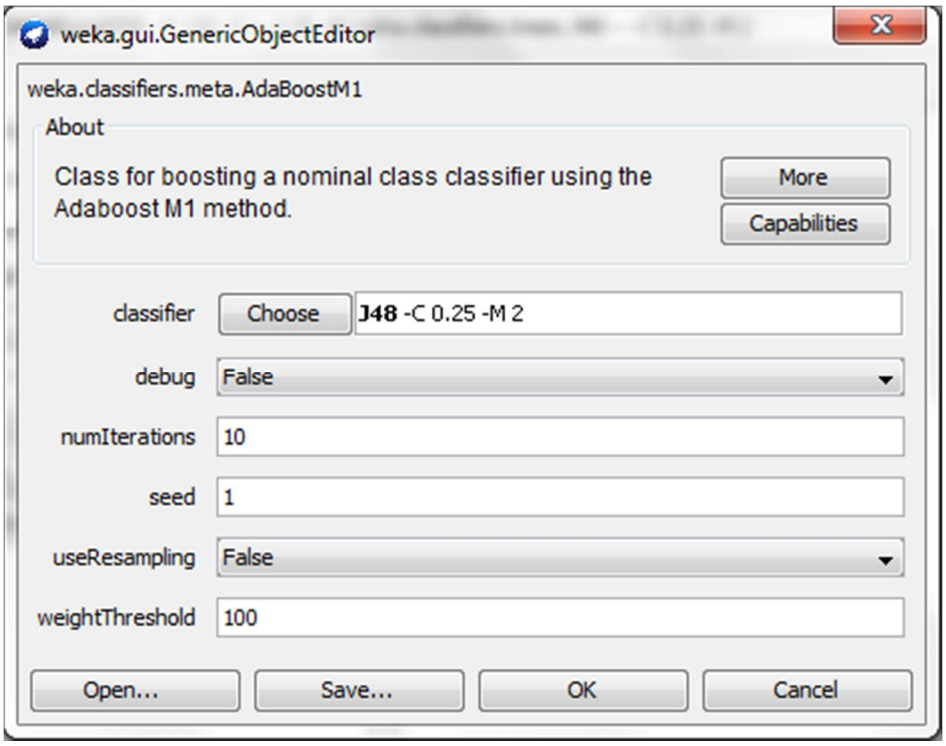
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	22 (0,063%)
	Negativas	0 (0%)	34.674 (99,937%)

Los resultados sobre las instancias positivas nos dan una idea de que este modelo quizás no sea el que mejor evalúe. En cualquier caso, este modelo ha clasificado correctamente como negativas el 99,937% (34.674 de 34.696) de las instancias que son negativas.

4.3.2.6. Modelo AdaBoostM1-J48

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 13: Configuración usada para la generación del modelo AdaBoostM1-J48



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M
2
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero Fichero de entrenamiento para Weka, base para la generación de modelos.

El modelo resultante es un árbol de 2.604 hojas y un tamaño total de 2.741 nodos. Una parte del árbol de decisión:

```
P25 = A
|   P32 = E: 0 (150.0)
|   P32 = I
|   |   P29 = I
|   |   |   P16 = A: 0 (1.0)
|   |   |   P16 = C: 1 (0.0)
|   |   |   P16 = D: 0 (1.0)
|   |   |   P16 = E: 0 (2.0)
|   |   |   P16 = F: 1 (133.0)
```

Este trozo del árbol parte de la condición que la posición 25 tenga valor *A*. A continuación evalúa el valor de la posición 32, y en función de su valor determina si la sub-secuencia es negativa (valor *E* -150 casos en el fichero de entrenamiento-), o si debe seguir evaluando la sub-secuencia (valor *I*). En este caso evalúa a continuación la posición 29, y en caso de que tenga el valor *I* pasa a evaluar la posición 16, que en función de sus valores determina si la sub-secuencia es positiva

(valores *C* –ningún caso- y *F* -133 casos-) o negativa (valores *A* -1 caso-, *D* -1 caso- y *E* -2 casos-).

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 8: Matriz de confusión de la generación del modelo AdaBoostM1-J48

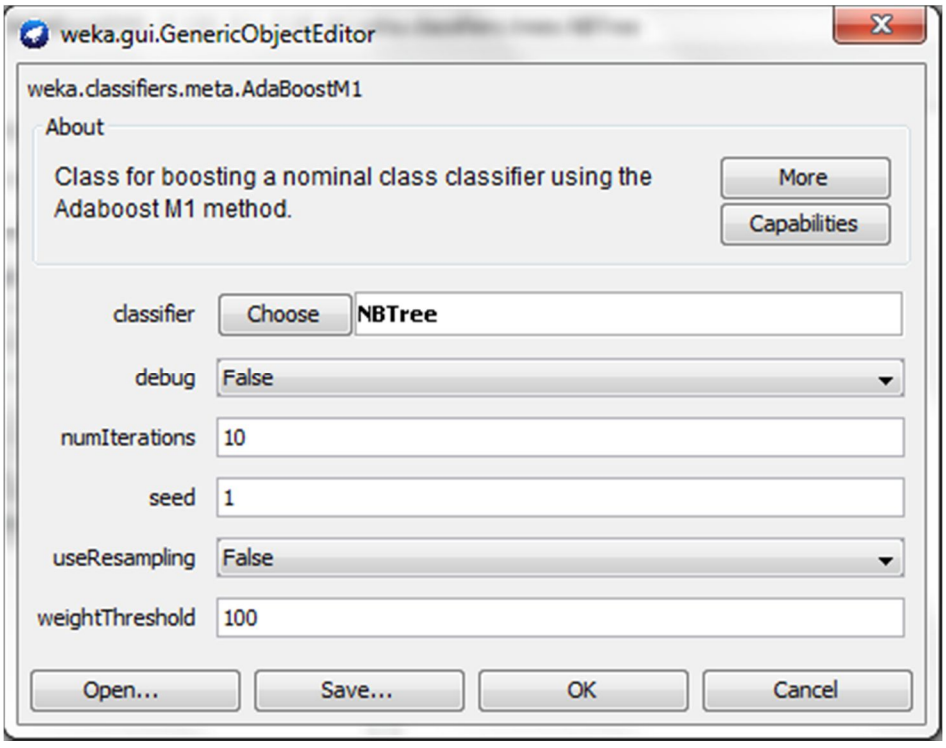
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	15 (0,043%)
	Negativas	0 (0%)	34.681 (99,957%)

Los resultados sobre las instancias positivas nos dan una idea de que este modelo quizás no sea el que mejor evalúe. En cualquier caso, este modelo ha clasificado correctamente como negativas el 99,957% (34.681 de 34.696) de las instancias que son negativas.

4.3.2.7. Modelo AdaBoostM1-NBTree

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 14: Configuración usada para la generación del modelo AdaBoostM1-NBTree



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.NBTree
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero Fichero de entrenamiento para Weka, base para la generación de modelos.

El modelo resultante es un árbol de 39 hojas (con un clasificador en cada una de ellas) y un tamaño total de 41 nodos. Algunas de las hojas:

```
P1 = A: NB 1
P1 = C: NB 2
P1 = D: NB 3
P1 = E: NB 4
P1 = F: NB 5
P1 = G
|   P4 = A: NB 7
```

Las hojas se definen en función del valor de la posición 1, y cuando ésta tiene valor *G* se evalúa la posición 4. Cada una de las hojas se compone de un clasificador compuesto por una tabla por cada uno de los posibles valores de cada una de las posiciones. Por ejemplo, la primera de las hojas (para cuando la posición 1 tiene valor *A*):

```
Leaf number: 1 Naive Bayes Classifier

               Class
Attribute          0      1
              (0.49) (0.51)
==============================
P1
  A             2473.0 2528.0
  C                1.0    1.0
  D                1.0    1.0
  E                1.0    1.0
  F                1.0    1.0
  G                1.0    1.0
  H                1.0    1.0
  I                1.0    1.0
  K                1.0    1.0
  L                1.0    1.0
  M                1.0    1.0
  N                1.0    1.0
  P                1.0    1.0
  Q                1.0    1.0
  R                1.0    1.0
  S                1.0    1.0
  T                1.0    1.0
  V                1.0    1.0
  W                1.0    1.0
  Y                1.0    1.0
  [total]       2492.0 2547.0

P2
  A              252.0  267.0
  C               39.0    1.0
  D              120.0  134.0
  E              159.0  134.0
  F               84.0    1.0
  G               94.0  533.0
  H               51.0    1.0
  I              149.0  134.0
  K              188.0  134.0
  L              303.0  134.0
  M               68.0  134.0
  N               88.0  134.0
  P              104.0    1.0
  Q               99.0    1.0
  R              120.0    1.0
  S              202.0  134.0
  T              128.0  400.0
  V              182.0  134.0
  W               12.0  134.0
  Y               50.0    1.0
  [total]       2492.0 2547.0

P3
  A              213.0    1.0
  C               46.0    1.0
  D              149.0  134.0
  E              177.0  267.0
```

```
F              106.0    1.0
G              118.0  267.0
H               54.0    1.0
I              128.0  134.0
K              171.0  267.0
L              282.0  267.0
M               59.0  134.0
N               98.0    1.0
P               93.0  267.0
Q              108.0  400.0
R              135.0    1.0
S              201.0  400.0
T              123.0    1.0
V              160.0    1.0
W               12.0    1.0
Y               59.0    1.0
[total]       2492.0 2547.0
```

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 9: Matriz de confusión de la generación del modelo AdaBoostM1-NBTree

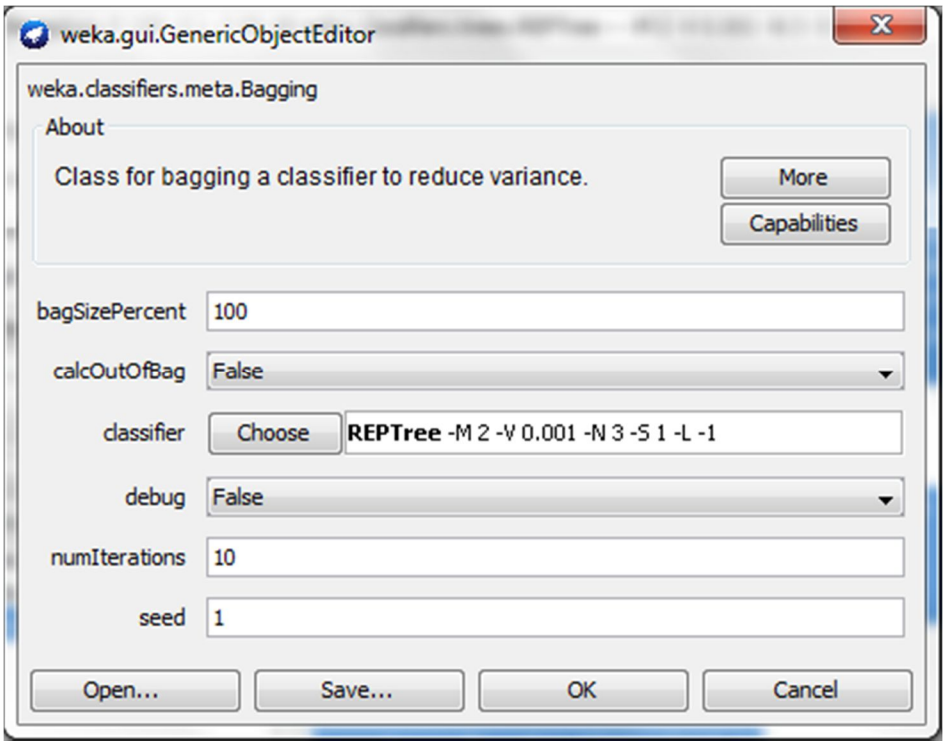
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	2 (0,006%)
	Negativas	0 (0%)	34.694 (99,994%)

Los resultados sobre las instancias positivas nos dan una idea de que este modelo quizás no sea el que mejor evalúe. En cualquier caso, este modelo ha clasificado correctamente como negativas el 99,994% (34.694 de 34.696) de las instancias que son negativas.

4.3.2.8. Modelo Bagging

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 15: Configuración usada para la generación del modelo Bagging



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.trees.REPTree -- -M 2 -V
0.001 -N 3 -S 1 -L -1
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero [Fichero de entrenamiento para Weka, base para la generación de modelos](#).

El árbol resultante se compone de 2.641 nodos. Una parte del árbol de decisión:

```
P25 = A
|   P32 = A
|   |   P7 = A : 0 (15/0) [7/0]
|   |   P7 = C : 0 (2/0) [2/0]
…
|   P32 = C : 0 (35/0) [15/0]
|   P32 = D : 0 (92/0) [44/0]
|   P32 = E : 0 (119/0) [39/0]
|   P32 = F
|   |   P36 = A : 0 (5/0) [1/0]
|   |   P36 = C : 1 (89/0) [37/0]
|   |   P36 = D : 0 (3/0) [0/0]
|   |   P36 = E : 0 (9/0) [3/0]
|   |   P36 = F : 0 (1/0) [1/0]
|   |   P36 = G : 1 (85/1) [41/2]
…
|   P32 = I
|   |   P11 = A : 0 (5/0) [4/0]
|   |   P11 = C : 0 (0/0) [3/0]
|   |   P11 = D : 0 (9/0) [2/0]
|   |   P11 = E : 0 (5/0) [3/0]
|   |   P11 = F : 0 (1/0) [0/0]
|   |   P11 = G
```

```
|   |   |   P1 = A : 0 (1/0) [0/0]
|   |   |   P1 = C : 1 (0/0) [0/0]
|   |   |   P1 = D : 1 (0/0) [0/0]
|   |   |   P1 = E : 1 (0/0) [0/0]
|   |   |   P1 = F : 1 (0/0) [0/0]
|   |   |   P1 = G : 1 (0/0) [0/0]
|   |   |   P1 = H : 1 (0/0) [0/0]
|   |   |   P1 = I : 1 (0/0) [0/0]
|   |   |   P1 = K : 1 (0/0) [0/0]
|   |   |   P1 = L : 0 (2/0) [1/0]
|   |   |   P1 = M : 1 (0/0) [0/0]
|   |   |   P1 = N : 1 (0/0) [0/0]
|   |   |   P1 = P : 0 (1/0) [0/0]
|   |   |   P1 = Q : 1 (0/0) [0/0]
|   |   |   P1 = R : 1 (0/0) [0/0]
|   |   |   P1 = S : 1 (91/0) [38/0]
```

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 10: Matriz de confusión de la generación del modelo Bagging

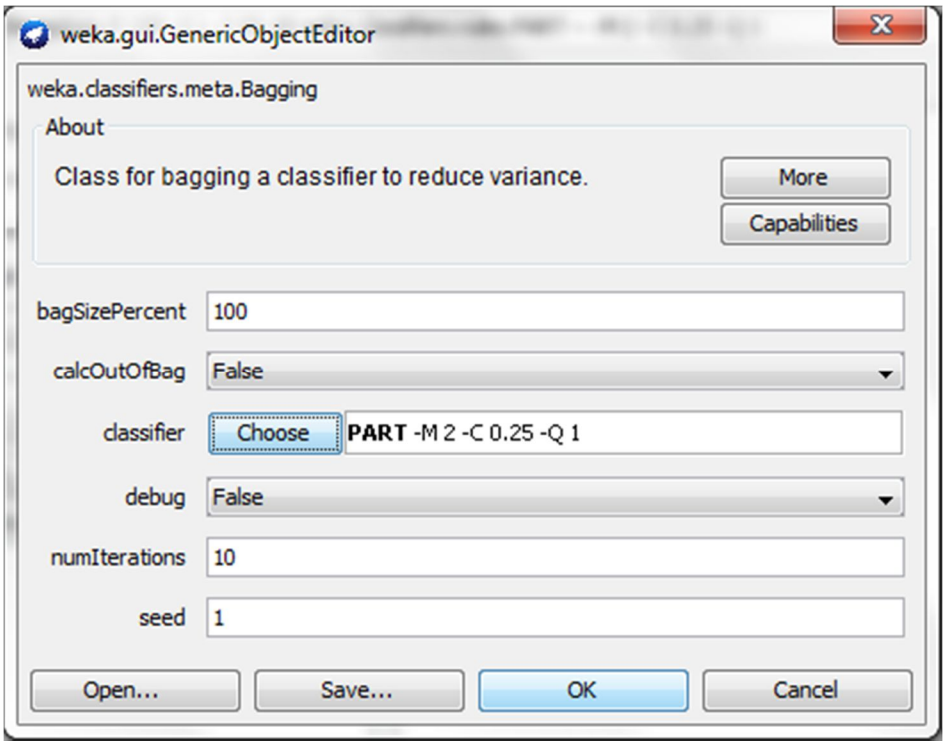
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	492 (1,418%)
	Negativas	0 (0%)	34.204 (98,582%)

Los resultados sobre las instancias positivas no son representativos ya que están repetidas, y por tanto se han utilizado ya en el proceso de aprendizaje y generación de reglas. El dato interesante de esta tabla es que este modelo ha clasificado correctamente como negativas el 98,582% (34.204 de 34.696) de las instancias que son negativas.

4.3.2.9. Modelo Bagging-Part

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 16: Configuración usada para la generación del modelo Bagging-Part



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25
-Q 1
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero [Fichero de entrenamiento para Weka, base para la generación de modelos](#).

El modelo resultante se compone de 163 reglas. Algunas de estas reglas:

```
P36 = E: 0 (652.0)
```

```
P25 = R AND
P13 = L AND
P17 = L AND
P16 = L: 1 (557.0)
```

```
P13 = L AND
P29 = S: 0 (166.0)
```

```
P13 = L AND
P19 = D AND
P16 = L AND
P20 = S: 1 (383.0/1.0)
```

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 11: Matriz de confusión de la generación del modelo Bagging-Part

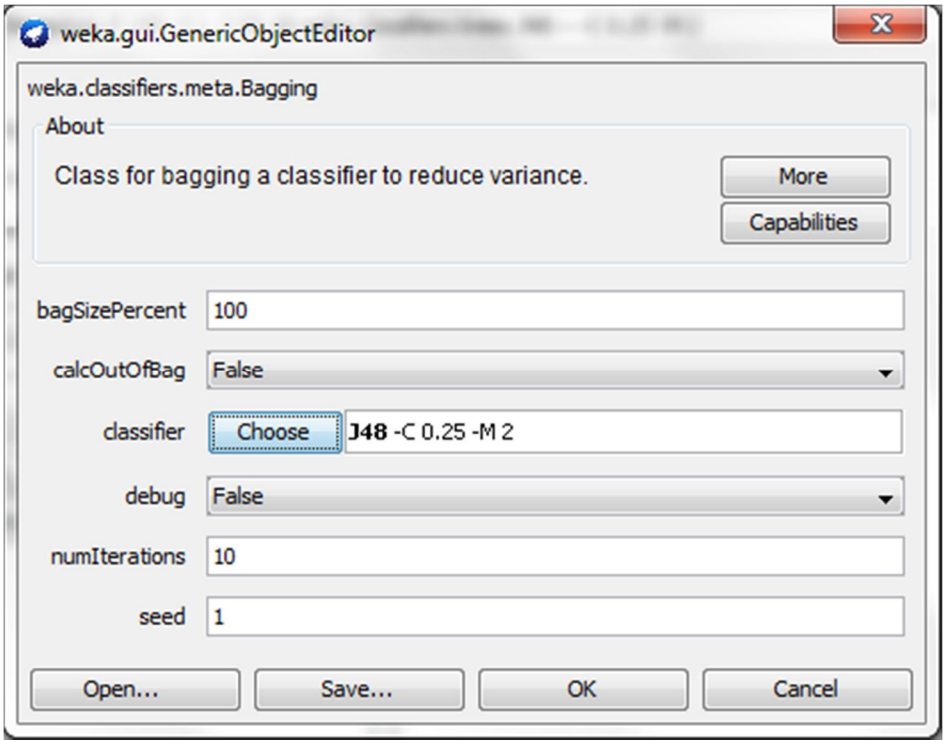
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	43 (0,124%)
	Negativas	0 (0%)	34.653 (99,876%)

Los resultados sobre las instancias positivas no son representativos ya que están repetidas, y por tanto se han utilizado ya en el proceso de aprendizaje y generación de reglas. El dato interesante de esta tabla es que este modelo ha clasificado correctamente como negativas el 99,876% (34.653 de 34.696) de las instancias que son negativas.

4.3.2.10. Modelo Bagging-J48

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 17: Configuración usada para la generación del modelo Bagging-J48



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero Fichero de entrenamiento para Weka, base para la generación de modelos.

El árbol resultante se compone de 3.193 hojas, con un tamaño total de 3.361 nodos. Una parte del árbol de decisión:

```
P25 = S
|   P24 = A: 0 (194.0)
|   P24 = C: 0 (35.0)
|   P24 = D: 0 (160.0)
|   P24 = E: 0 (181.0)
|   P24 = F: 0 (126.0)
|   P24 = G: 0 (76.0)
|   P24 = H: 0 (41.0)
|   P24 = I
|   |   P13 = A: 0 (14.0)
|   |   P13 = C: 1 (0.0)
|   |   P13 = D: 0 (6.0)
|   |   P13 = E: 0 (11.0)
|   |   P13 = F: 0 (16.0)
|   |   P13 = G: 0 (3.0)
|   |   P13 = H: 0 (4.0)
|   |   P13 = I: 0 (14.0)
|   |   P13 = K: 0 (11.0)
|   |   P13 = L
|   |   |   P7 = A: 1 (0.0)
|   |   |   P7 = C: 1 (0.0)
|   |   |   P7 = D: 1 (0.0)
|   |   |   P7 = E: 1 (0.0)
|   |   |   P7 = F: 1 (133.0)
|   |   |   P7 = G: 1 (0.0)
```

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 12: Matriz de confusión de la generación del modelo Bagging-J48

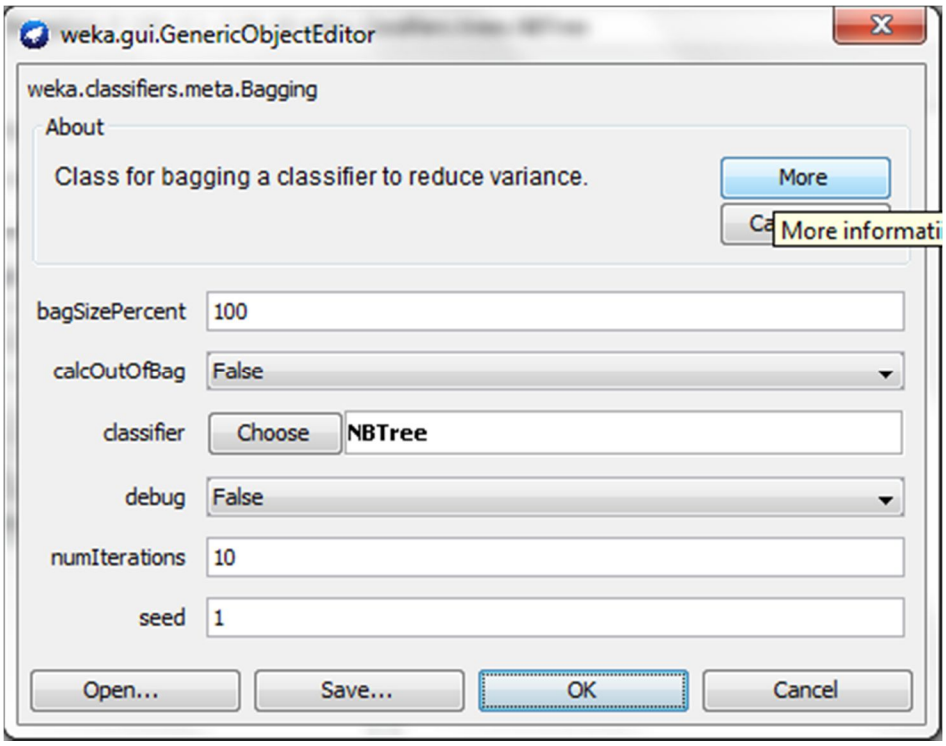
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	672 (1,937%)
	Negativas	0 (0%)	34.024 (98,063%)

Los resultados sobre las instancias positivas no son representativos ya que están repetidas, y por tanto se han utilizado ya en el proceso de aprendizaje y generación de reglas. El dato interesante de esta tabla es que este modelo ha clasificado correctamente como negativas el 98,063% (34.024 de 34.696) de las instancias que son negativas.

4.3.2.11. Modelo Bagging-NBTree

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 18: Configuración usada para la generación del modelo Bagging-NBTree



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.trees.NBTree
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero Fichero de entrenamiento para Weka, base para la generación de modelos.

El árbol resultante se compone de 20 hojas (con un clasificador en cada una de ellas), con un tamaño total de 21 nodos. Algunas de las hojas:

```
P1 = A: NB 1
P1 = C: NB 2
P1 = D: NB 3
P1 = E: NB 4
P1 = F: NB 5
```

Las hojas se definen en función del valor de la posición 1. Cada una de las hojas se compone de un clasificador compuesto por una tabla por cada uno de los posibles valores de cada una de las posiciones. Por ejemplo, la primera de las hojas (para cuando la posición 1 tiene valor *A*):

```
Leaf number: 1 Naive Bayes Classifier

                Class
Attribute           0      1
                (0.5)  (0.5)
============================
P1
  A            2544.0 2534.0
```

```
  C                1.0    1.0
  D                1.0    1.0
  E                1.0    1.0
  F                1.0    1.0
  G                1.0    1.0
  H                1.0    1.0
  I                1.0    1.0
  K                1.0    1.0
  L                1.0    1.0
  M                1.0    1.0
  N                1.0    1.0
  P                1.0    1.0
  Q                1.0    1.0
  R                1.0    1.0
  S                1.0    1.0
  T                1.0    1.0
  V                1.0    1.0
  W                1.0    1.0
  Y                1.0    1.0
  [total]       2563.0 2553.0

P2
  A              255.0  234.0
  C               43.0    1.0
  D              125.0  157.0
  E              155.0  147.0
  F               87.0    1.0
  G               97.0  529.0
  H               45.0    1.0
  I              154.0  155.0
  K              187.0  127.0
  L              308.0  123.0
  M               74.0  123.0
  N              104.0  156.0
  P              109.0    1.0
  Q              105.0    1.0
  R              117.0    1.0
  S              200.0  132.0
  T              124.0  390.0
  V              196.0  136.0
  W               12.0  137.0
  Y               66.0    1.0
  [total]       2563.0 2553.0

P3
  A              209.0    1.0
  C               60.0    1.0
  D              156.0  123.0
  E              188.0  310.0
  F              123.0    1.0
  G              114.0  283.0
  H               61.0    1.0
  I              119.0  112.0
  K              175.0  256.0
  L              302.0  264.0
  M               67.0  155.0
  N              101.0    1.0
  P               90.0  237.0
  Q               94.0  415.0
  R              145.0    1.0
  S              205.0  388.0
  T              124.0    1.0
  V              170.0    1.0
  W               10.0    1.0
  Y               50.0    1.0
  [total]       2563.0 2553.0
…
```

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 13: Matriz de confusión de la generación del modelo Bagging-NBTree

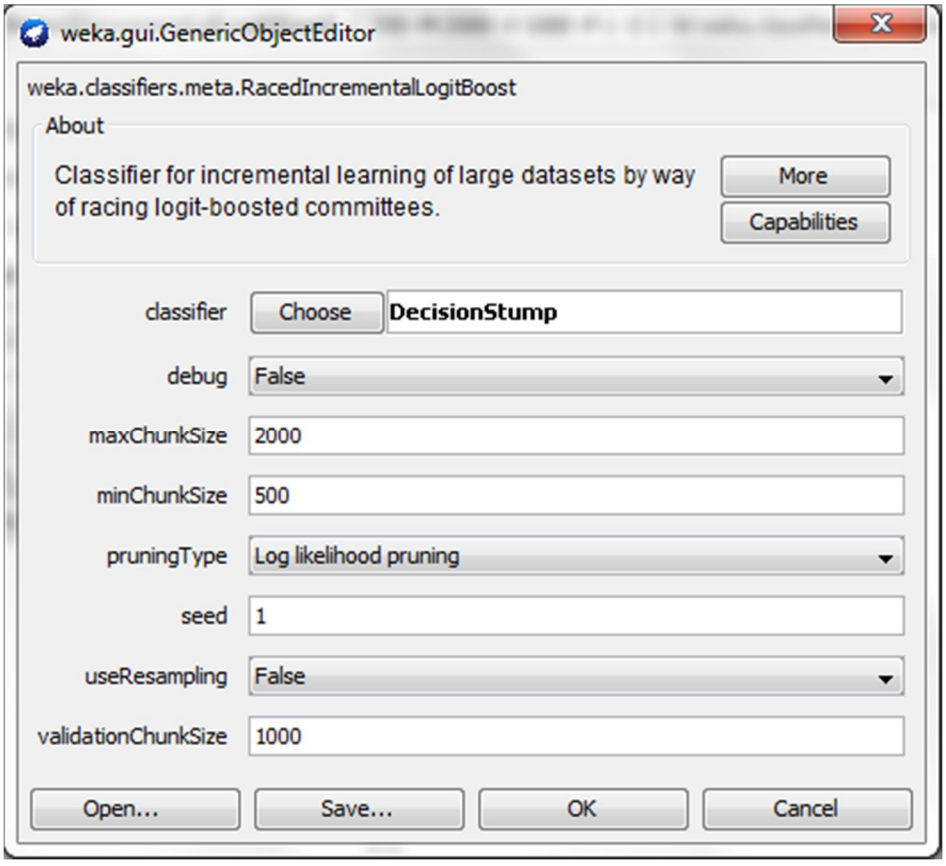
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	1 (0,003%)
	Negativas	0 (0%)	34.695 (99,997%)

Los resultados sobre las instancias positivas no son representativos ya que están repetidas, y por tanto se han utilizado ya en el proceso de aprendizaje y generación de reglas. El dato interesante de esta tabla es que este modelo ha clasificado correctamente como negativas el 99,997% (34.695 de 34.696) de las instancias que son negativas.

4.3.2.12. Modelo RacedIncrementalLogitBoost

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 19: Configuración usada para la generación del modelo RacedIncrementalLogitBoost



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.RacedIncrementalLogitBoost -C 500 -M 2000 -V 1000 -P 1 -S 1 -W
weka.classifiers.trees.DecisionStump
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero Fichero de entrenamiento para Weka, base para la generación de modelos.

Este modelo está compuesto por 56 sub-modelos. Uno de estos sub-modelos:

```
Model 7
       Class 1 (CLASS=0)

Decision Stump

Classifications

P13 = I : -1.6294714670672386
P13 != I : 0.17262991532369878
P13 is missing : -0.09767992249326406


       Class 2 (CLASS=1)

Decision Stump

Classifications

P13 = I : 1.6294714670672388
P13 != I : -0.17262991532369903
P13 is missing : 0.097679922493264
```

Este sub-modelo evalúa la posición 13 de la sub-secuencia relacionando el peso que tiene cuando dicha posición tiene valor *I* u otro valor distinto, para determinar si la sub-secuencia es positiva o negativa.

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 14: Matriz de confusión de la generación del modelo RacedIncrementalLogitBoost

		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	33.026 (95,506%)	2.854 (8,226%)
	Negativas	1.554 (4,494%)	31.842 (91,774%)

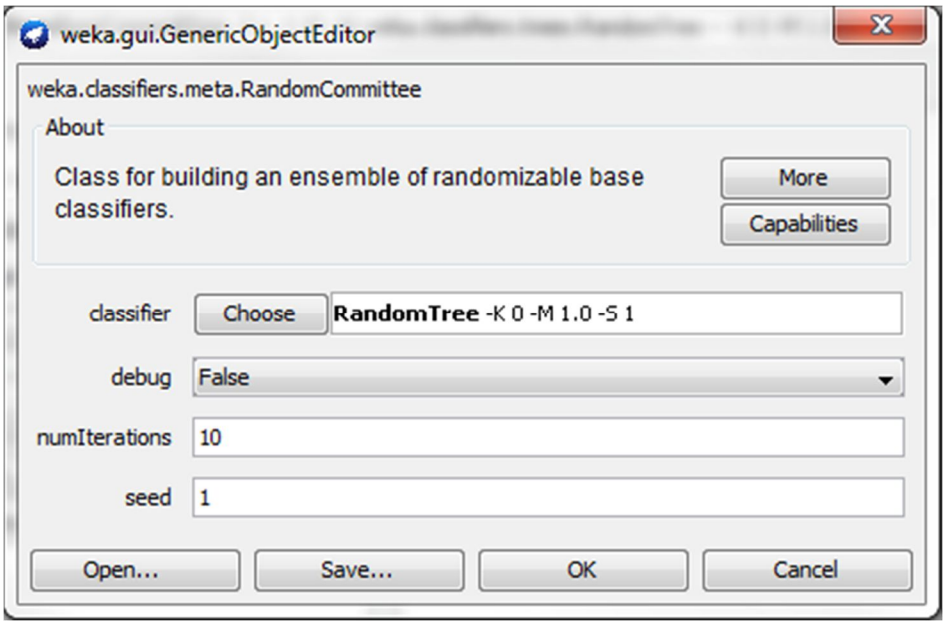
Los resultados sobre las instancias positivas nos dan una idea de que este modelo quizás no sea el que mejor evalúe, ya que dichas instancias están repetidas, por lo que es más sencillo poder catalogarlas. En cualquier caso, este modelo ha clasificado correctamente como negativas el 91,774% (31.842 de 34.696) de las instancias que son negativas.

El clasificador RacedIncrementalLogitBoost no admite ser combinado con los otros clasificadores que estamos usando (Part, J48 y NBTree).

4.3.2.13. Modelo RandomCommittee

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 20: Configuración usada para la generación del modelo RandomCommittee



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.RandomCommittee -S 1 -I 10 -W weka.classifiers.trees.RandomTree -- -K 0 -M 1.0 -S 1
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero [Fichero de entrenamiento para Weka, base para la generación de modelos].

El árbol resultante tiene un tamaño total de 6.341 nodos. Parte del árbol de decisión:

```
P13 = C
|   P26 = A : 0 (33/0)
|   P26 = C : 0 (10/0)
|   P26 = D : 0 (31/0)
|   P26 = E : 0 (39/0)
|   P26 = F
|   |   P14 = A : 0 (0/0)
|   |   P14 = C : 1 (133/0)
…
P13 = H
|   P1 = A : 0 (48/0)
|   P1 = C : 0 (10/0)
|   P1 = D : 0 (33/0)
|   P1 = E : 0 (52/0)
|   P1 = F : 0 (33/0)
|   P1 = G
|   |   P20 = A : 0 (5/0)
|   |   P20 = C : 0 (0/0)
|   |   P20 = D : 1 (133/0)
```

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 15: Matriz de confusión de la generación del modelo RandomCommittee

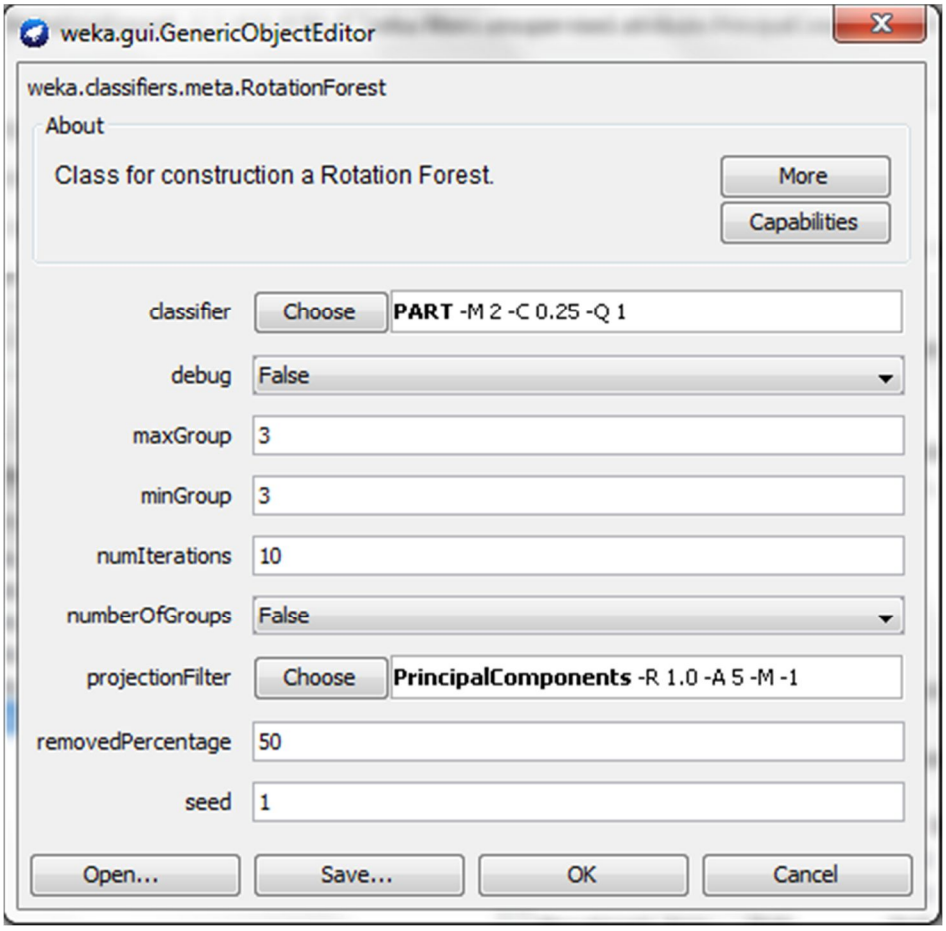
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	0 (0%)
	Negativas	0 (0%)	34.696 (100%)

Los resultados sobre las instancias positivas no son representativos ya que están repetidas, y por tanto se han utilizado ya en el proceso de aprendizaje y generación de reglas. El dato interesante de esta tabla es que este modelo ha clasificado correctamente como negativas el 100% de las instancias que son negativas.

4.3.2.14. Modelo RotationForest-Part

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 21: Configuración usada para la generación del modelo RotationForest-Part



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.RotationForest -G 3 -H 3 -P 50 -F
"weka.filters.unsupervised.attribute.PrincipalComponents -R 1.0 -A 5 -M -1" -S 1 -I 10 -W
weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero [Fichero de entrenamiento para Weka, base para la generación de modelos].

El modelo resultante consta de 18 reglas. Un par de estas reglas:

```
-0.413P36_0=D+0.376P37_1=W+0.297P36_0=Q+0.296P12_2=Y-0.246P12_2=F..._9 > -0.678519 AND
0.328P23_1=K-0.308P20_0=K-0.296P33_2=W+0.26 P23_1=Q-0.244P23_1=M..._12 <= 0.673212 AND
0.413P12_2=P-0.277P37_1=Y+0.253P12_2=V-0.231P36_0=H-0.22P12_2=H..._9 <= 0.474224 AND
0.419P2_0=T-0.403P11_1=W+0.334P11_1=K-0.242P29_2=Q-0.194P2_0=Y..._0 > -1.002524 AND
-0.327P33_2=Y-0.319P23_1=G+0.267P20_0=G+0.257P23_1=S+0.235P23_1=R..._12 <= 0.817 AND
-0.277P15_2=D+0.274P25_1=R+0.236P15_2=T-0.21P25_1=Q+0.203P15_2=A..._5 > -1.037413 AND
0.316P9_1=P-0.292P30_2=F-0.281P30_2=S+0.261P9_1=Q+0.258P16_0=V..._1 <= 0.91548 AND
0.271P25_1=T+0.26 P15_2=T+0.257P15_2=C+0.247P21_0=E+0.237P21_0=Y..._5 <= 0.701575 AND
0.324P22_2=Q-0.274P14_0=G-0.253P14_0=M+0.252P14_0=Y+0.243P22_2=V..._11 <= 0.441944 AND
-0.289P38_0=R-0.261P34_2=S+0.246P19_1=E-0.246P19_1=N-0.245P19_1=T..._8 <= 0.785664 AND
0.347P27_2=Q+0.332P5_0=Q-0.329P27_2=S+0.279P5_0=Y-0.245P5_0=A..._2 > -0.880552 AND
0.388P36_0=T-0.261P36_0=Q+0.24  P12_2=I+0.235P36_0=H+0.214P36_0=V..._9  <=  1.774797:  1
(3598.0/7.0)
```

```
0.386P4_0=P-0.304P7_1=E+0.287P7_1=R-0.248P7_1=N+0.233P32_2=H..._4 <= 3.214019 AND
0.527P10_2=I+0.396P8_1=P+0.301P3_0=P-0.199P3_0=T+0.193P3_0=E..._7 <= 2.52902 AND
-0.48P10_2=V-0.318P8_1=E+0.271P8_1=I+0.225P8_1=V+0.216P10_2=N..._7 > -3.174519 AND
-0.284P6_1=Q+0.251P5_0=V-0.251P5_0=S+0.244P5_0=T-0.216P5_0=G..._2 > -4.576156 AND
0.527P10_2=L+0.292P3_0=S+0.267P8_1=D-0.257P8_1=V-0.203P3_0=T..._7 > -3.598162 AND
0.545P26_1=R+0.367P18_0=L-0.275P17_2=L+0.219P17_2=H+0.219P17_2=C..._3 > -1.77013 AND
0.299P12_2=M+0.292P36_0=W+0.284P36_0=G-0.278P12_2=H+0.25 P37_1=I..._9 <= 5.187888 AND
0.319P22_2=R-0.313P22_2=W-0.296P35_1=S-0.264P35_1=I+0.247P35_1=H..._11 > -3.667707 AND
0.311P37_1=T+0.225P12_2=E-0.223P36_0=I-0.22P12_2=Y+0.216P12_2=W..._9 <= 4.052545 AND
-0.33P24_0=A-0.311P28_1=A+0.285P39_2=F+0.28 P24_0=N+0.237P24_0=K..._10 > -2.903857 AND
-0.347P32_2=C+0.346P32_2=A-0.316P4_0=H+0.307P7_1=S+0.239P4_0=Q..._4 <= 2.50868 AND
0.271P27_2=C-0.246P27_2=S-0.245P5_0=D-0.244P6_1=H+0.234P6_1=R..._2 > -3.268538 AND
-0.318P24_0=G+0.265P28_1=N-0.253P28_1=V+0.235P39_2=S-0.225P39_2=P..._10   >   -3.528362
AND
0.315P21_0=V+0.297P21_0=D+0.276P25_1=R-0.264P21_0=Q-0.223P25_1=H..._5 <= 3.296122 AND
0.306P13_2=P-0.284P13_2=N-0.276P1_0=G-0.255P31_1=C-0.243P1_0=M..._6 > -3.211433 AND
0.271P25_1=T+0.26  P15_2=T+0.257P15_2=C+0.247P21_0=E+0.237P21_0=Y..._5  >  -2.511667:  0
(33660.0)
```

En este modelo tenemos pocas reglas pero bastante más complejas ya que se trata de reglas de múltiples condiciones (desde 3 hasta 88) en las que se evalúan el valor de algunas posiciones y otorgándoles un peso a cada una de esas comparaciones, sumando/restando el resultado de cada peso y comparándolo con valores constantes.

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 16: Matriz de confusión de la generación del modelo RotationForest-Part

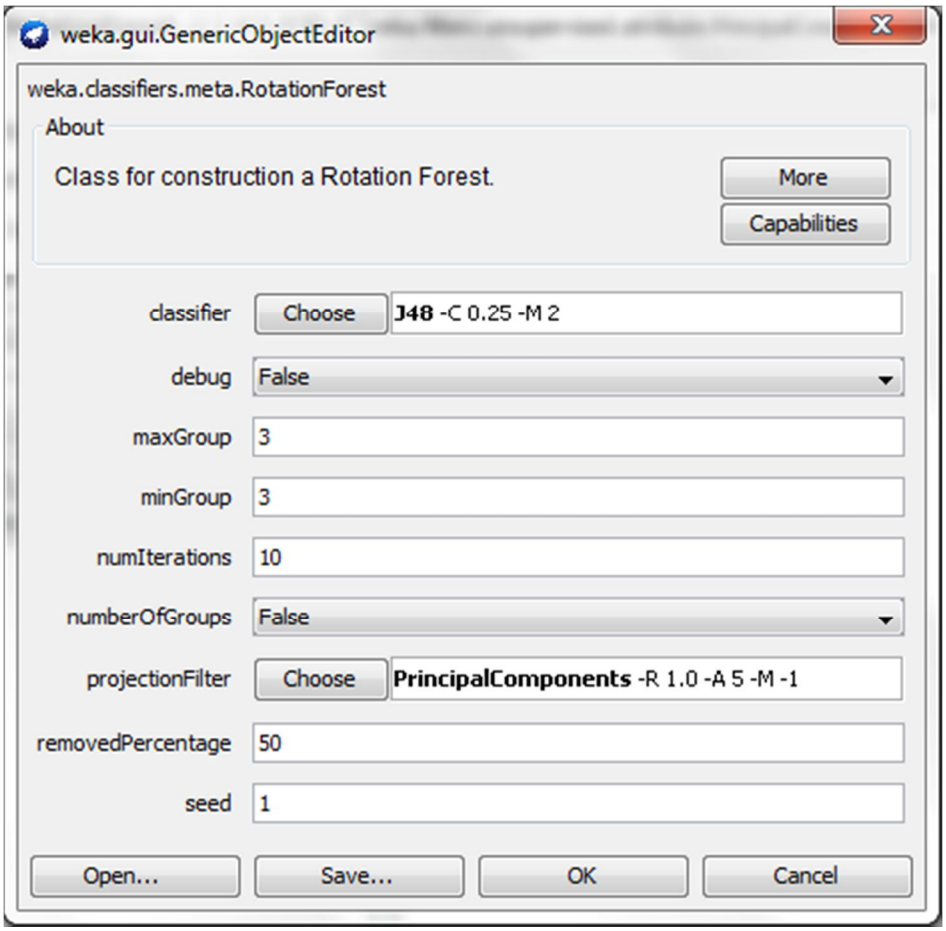
		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	0 (0%)
	Negativas	0 (0%)	34.696 (100%)

Los resultados sobre las instancias positivas no son representativos ya que están repetidas, y por tanto se han utilizado ya en el proceso de aprendizaje y generación de reglas. El dato interesante de esta tabla es que este modelo ha clasificado correctamente como negativas el 100% de las instancias que son negativas.

4.3.2.15. Modelo RotationForest-J48

El modelo se ha generado con los siguientes parámetros de configuración:

Figura 22: Configuración usada para la generación del modelo RotationForest-J48



Esta configuración se corresponde al comando:

```
weka.classifiers.meta.RotationForest      -G      3      -H      3      -P      50      -F
"weka.filters.unsupervised.attribute.PrincipalComponents  -R  1.0  -A  5  -M  -1"  -S  1  -I  10  -W
weka.classifiers.trees.J48 -- -C 0.25 -M 2
```

El modelo parte de las 69276 instancias (entre positivas y negativas) del fichero [Fichero de entrenamiento para Weka, base para la generación de modelos].

Este modelo genera 10 árboles de decisión podados. Por ejemplo, el último de ellos tiene 107 hojas y un tamaño total de 213 nodos, y una parte de dicho árbol:

```
-0.298P23_0=Q+0.268P22_2=L+0.259P16_1=N+0.237P16_1=Y-0.231P16_1=E..._2 <= 1.603084
|   -0.537P21_1=R+0.482P36_2=D+0.211P36_2=E-0.208P36_2=N+0.197P21_1=T..._6 <= 2.508037
|      |      -0.363P27_1=E+0.253P27_1=D+0.247P10_2=C+0.232P27_1=Q-0.214P25_0=D..._7  <=
1.820007
|   |   |   -0.341P31_0=L-0.331P31_0=A-0.311P31_0=K-0.31P31_0=E-0.292P31_0=I..._6 <= 0
|   |   |   |   0.284P7_2=M-0.247P33_1=D-0.244P12_0=V-0.226P7_2=A-0.224P7_2=L..._3 <=
-1.529344: 0 (1877.0)
|   |   |   |   0.284P7_2=M-0.247P33_1=D-0.244P12_0=V-0.226P7_2=A-0.224P7_2=L..._3 > -
1.529344
|      |      |      |      |      0.397P21_1=Q-0.359P36_2=K+0.277P36_2=G-0.238P31_0=S-
0.234P31_0=Q..._6 <= -1.650254
```

```
|     |     |     |     |     |     -0.583P19_2=D-0.537P24_0=V-0.205P3_1=N-0.174P3_1=P-
0.157P24_0=I..._0 <= -2.32429
|     |     |     |     |     |     0.393P33_1=L+0.318P7_2=E-0.257P33_1=G-0.252P7_2=P-
0.232P12_0=Q..._3 <= -1.385096: 1 (266.0)
|     |     |     |     |     |     0.393P33_1=L+0.318P7_2=E-0.257P33_1=G-0.252P7_2=P-
0.232P12_0=Q..._3 > -1.385096: 0 (22.0)
|     |     |     |     |     |     -0.583P19_2=D-0.537P24_0=V-0.205P3_1=N-0.174P3_1=P-
0.157P24_0=I..._0 > -2.32429: 0 (4010.0)
|     |     |     |     |     0.397P21_1=Q-0.359P36_2=K+0.277P36_2=G-0.238P31_0=S-
0.234P31_0=Q..._6 > -1.650254
|     |     |     |     |     |     -0.5P21_1=L+0.343P36_2=N+0.313P36_2=M+0.262P31_0=H-
0.246P31_0=R..._6 <= 1.792214
|     |     |     |     |     |     -0.363P27_1=E+0.253P27_1=D+0.247P10_2=C+0.232P27_1=Q-
0.214P25_0=D..._7 <= -2.006847: 0 (1147.0)
|     |     |     |     |     |     -0.363P27_1=E+0.253P27_1=D+0.247P10_2=C+0.232P27_1=Q-
0.214P25_0=D..._7 > -2.006847
…
```

Cada uno de los árboles generados se compone de nodos con reglas complejas en las que se evalúa el valor de ciertas posiciones asignándole un peso a cada una de las condiciones que se van cumpliendo y finalmente comparando el resultado a final a un valor constante.

Este modelo clasificado las instancias del fichero inicial de la siguiente manera:

Tabla 17: Matriz de confusión de la generación del modelo RotationForest-J48

		Proteínas que sabemos que son:	
		Positivas	Negativas
Clasificadas como:	Positivas	34.580 (100%)	0 (0%)
	Negativas	0 (0%)	34.696 (100%)

Los resultados sobre las instancias positivas no son representativos ya que están repetidas, y por tanto se han utilizado ya en el proceso de aprendizaje y generación de reglas. El dato interesante de esta tabla es que este modelo ha clasificado correctamente como negativas el 100% de las instancias que son negativas.

4.4. Aplicar los modelos sobre los ficheros de proteínas positivas y negativas

Una vez que tenemos los modelos generados con el fichero de entrenamiento es hora de aplicarlos sobre el fichero de proteínas positivas (para poder ponerlos a prueba sabiendo de antemano que todas las proteínas de entrada son positivas) y sobre el de proteínas negativas (para ver cuántas de las inicialmente catalogadas como negativas clasifica como positivas).

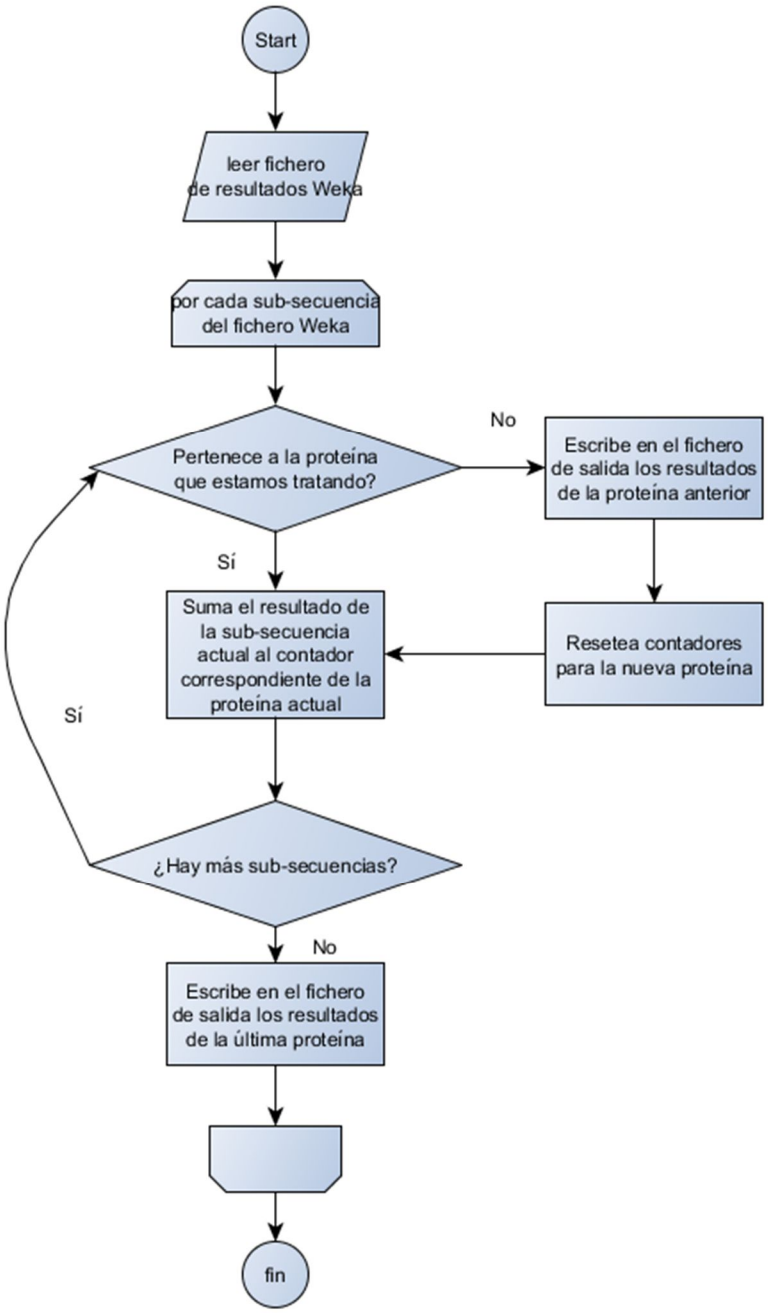
Hay que tener en cuenta que el fichero de proteínas positivas sobre el que vamos a actuar es tremendamente pequeño (129 proteínas) en comparación con el fichero de proteínas negativas (19.647 proteínas).

El fichero de proteínas positivas y el de proteínas negativas está compuesto de manera tal que una instancia se corresponde a una cadena de 39 aminoácidos de una proteína. Por tanto, una proteína está representada por 1 o varias entradas (instancias) en el fichero. Como los modelos se aplican sobre las instancias, al igual que se han generado a partir de las instancias, es necesario agrupar todas las instancias de cada una de las proteínas y ver cuántas instancias de cada proteína de ellas se han clasificado como positivas/negativas.

4.4.1. Agrupar los resultados por proteína

Para poder agrupar las sub-secuencias de aminoácidos en proteínas y así poder obtener la cantidad de sub-cadenas clasificadas como positivas y negativas en cada proteína se ha preparado un programa, cuya funcionalidad se resume en este diagrama UML de actividad:

Figura 23: Diagrama UML de actividad del programa para agrupar los resultados de las sub-secuencias de cada proteína, una vez que se ha aplicado un modelo sobre un fichero en Weka



4.4.2. Aplicación de los modelos generados sobre el fichero de proteínas positivas

A continuación se va a poner un resumen de los datos que resultan de aplicar cada uno de los modelos anteriormente generados, sobre el fichero de proteínas positivas.

Por cada uno de los modelos se incluyen unos gráficos que pretenden mostrar visualmente la clasificación de las proteínas. Para entenderlo mejor, los gráficos son:

- Un gráfico que indica cuántas proteínas hay con un número determinado de sub-secuencias de aminoácidos clasificadas como positivas.

- Un gráfico que indica el número de proteínas que hay en unos rangos de porcentaje calculados como relación de sub-secuencias de aminoácidos positivas sobre el total de sub-secuencias de la proteína. O dicho de otro modo, se refiere a un gráfico en el que se cuántas proteínas tienen en un X% y un Y% de sub-secuencias clasificadas como positivas.

4.4.2.1. Modelo Part

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 24: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo Part

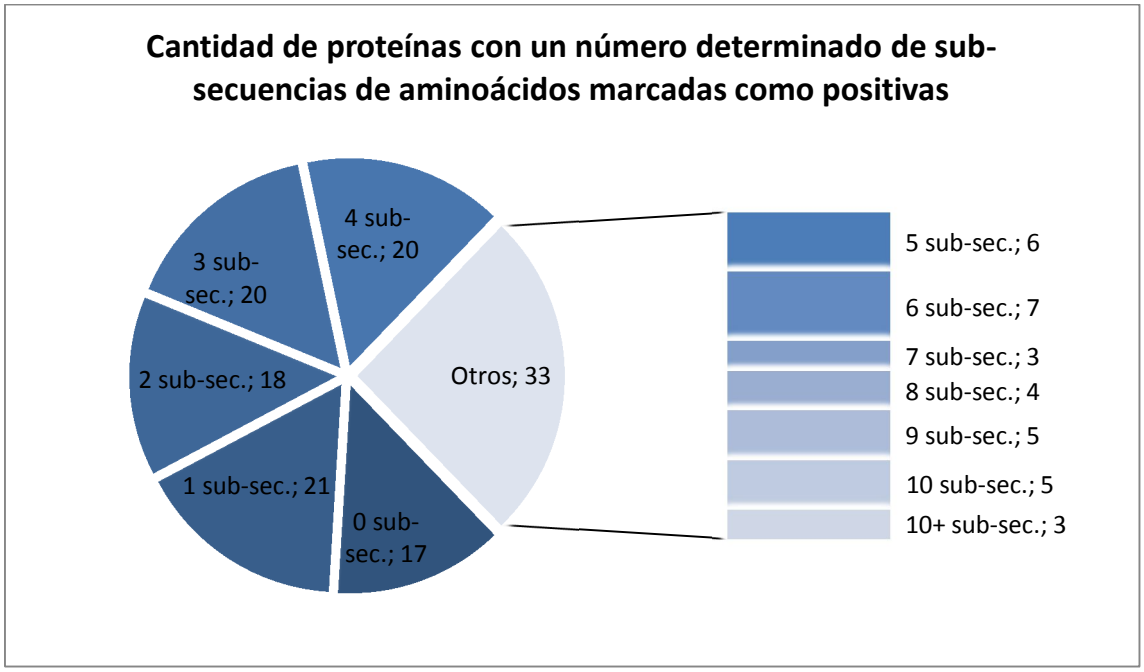
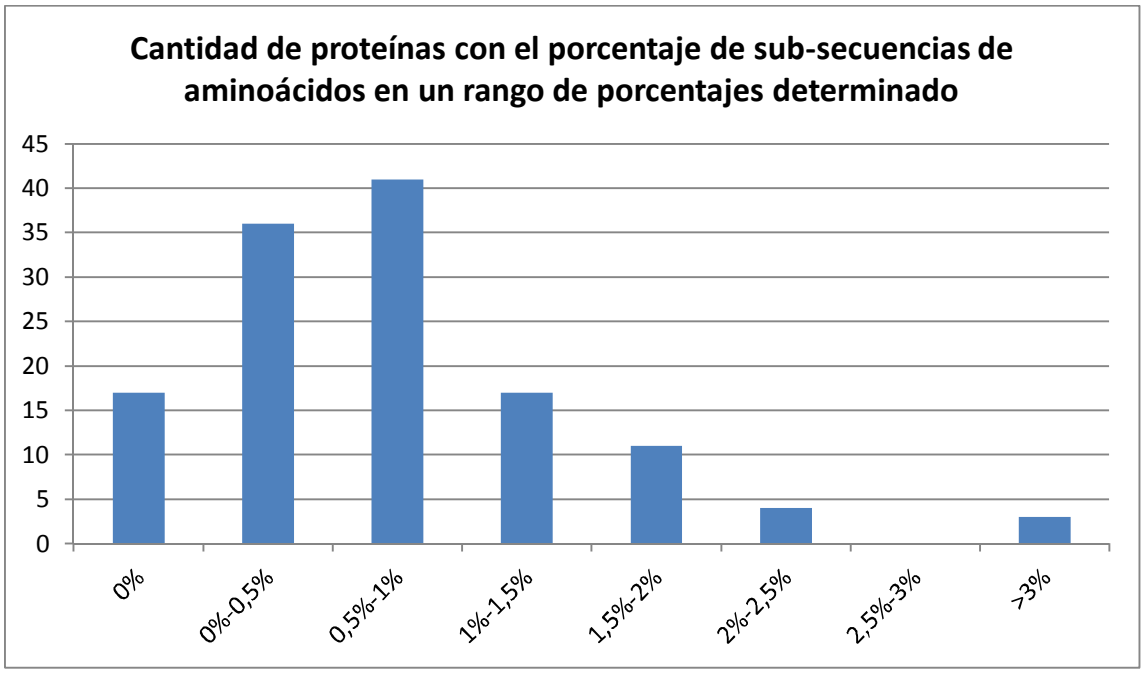


Figura 25: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado



4.4.2.2. Modelo J48

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 26: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo J48

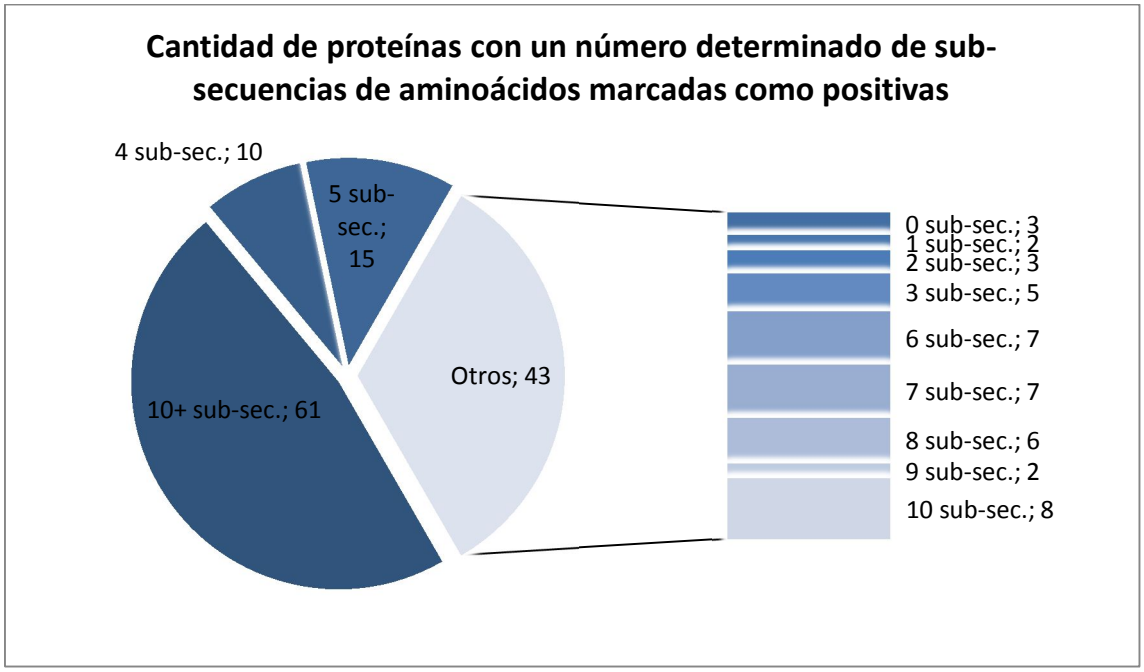
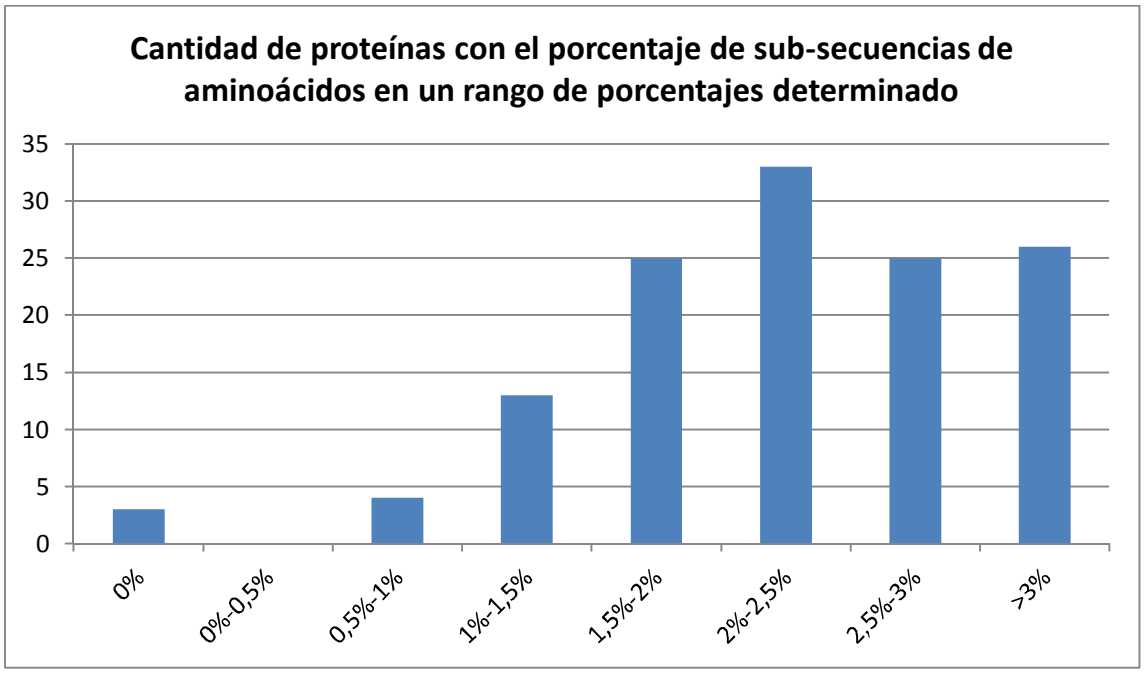


Figura 27: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado



4.4.2.3. Modelo NBTree

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 28: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo NBTree

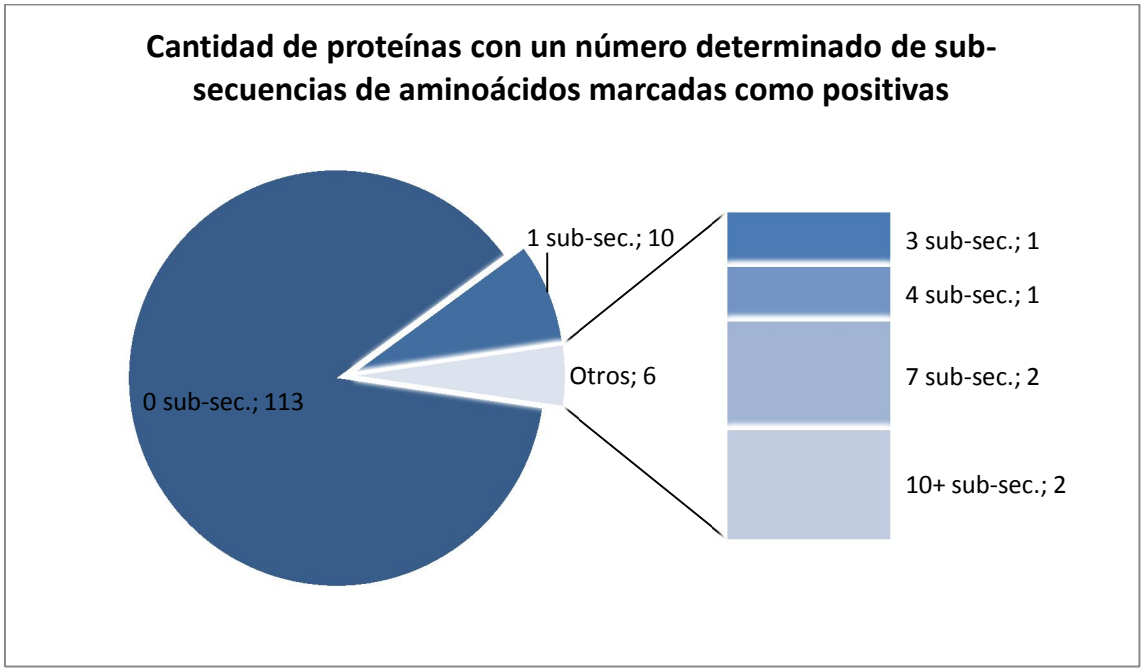
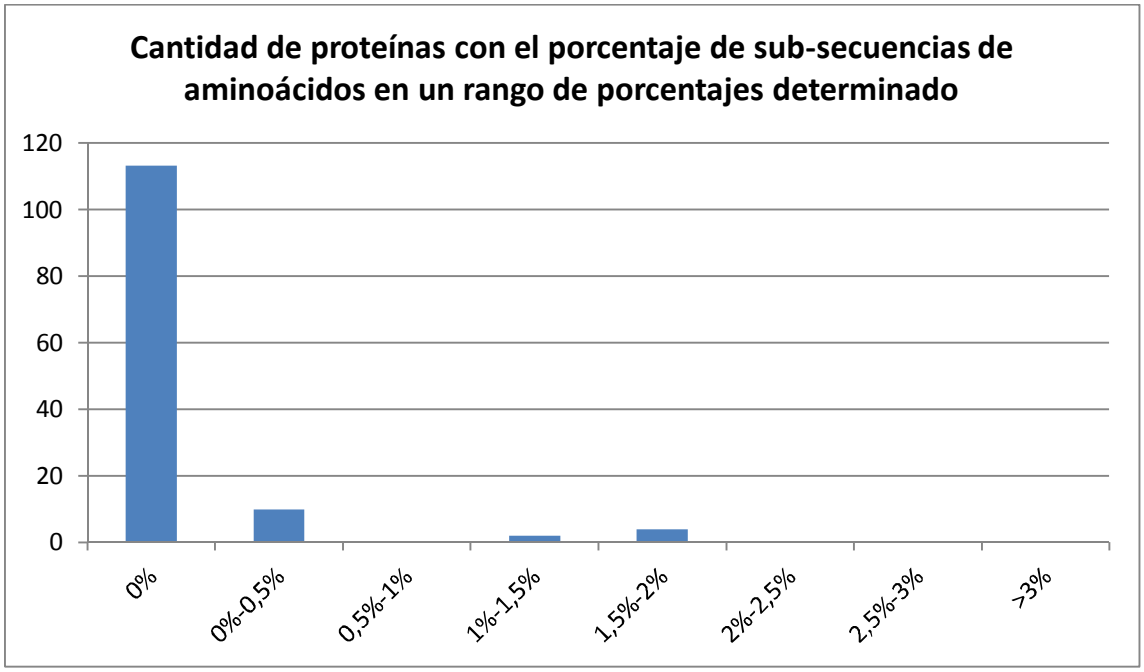


Figura 29: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo NBTree



4.4.2.4. Modelo AdaBoostM1

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 30: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo AdaBoostM1

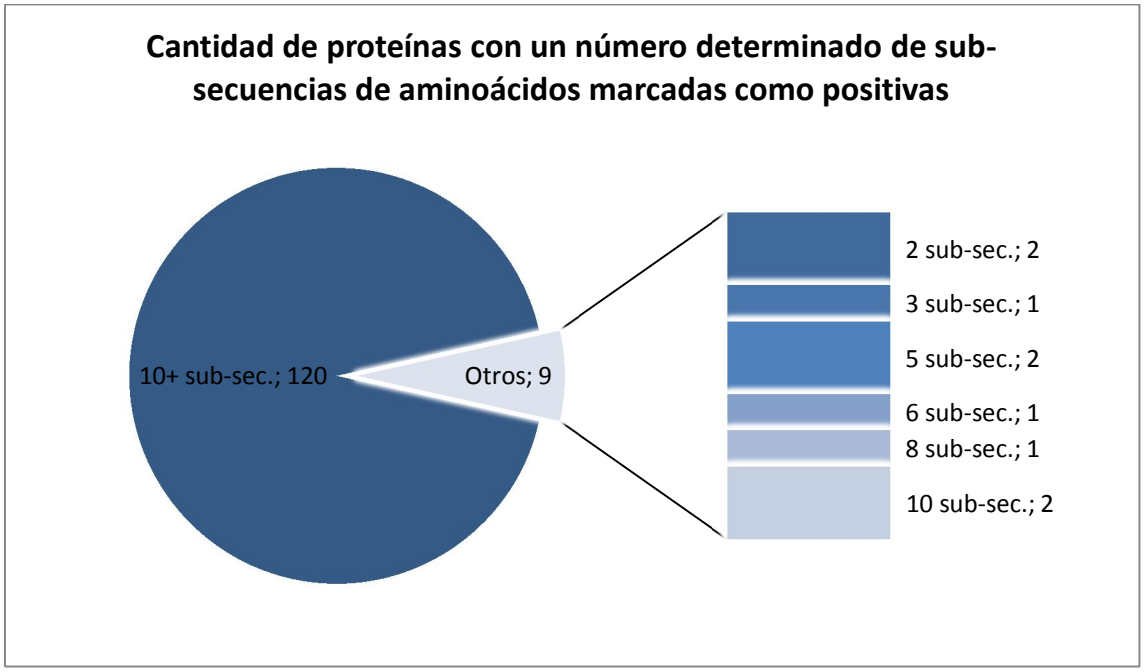
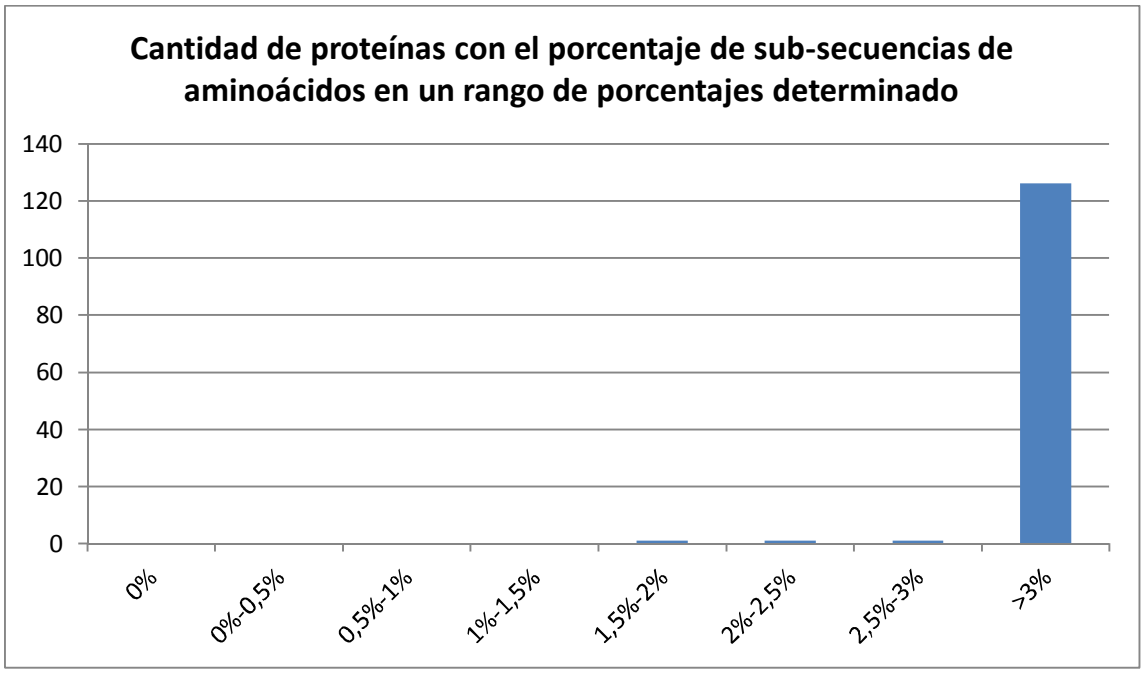


Figura 31: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo AdaBoostM1



4.4.2.5. Modelo AdaBoostM1-Part

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 32: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo AdaBoostM1-Part

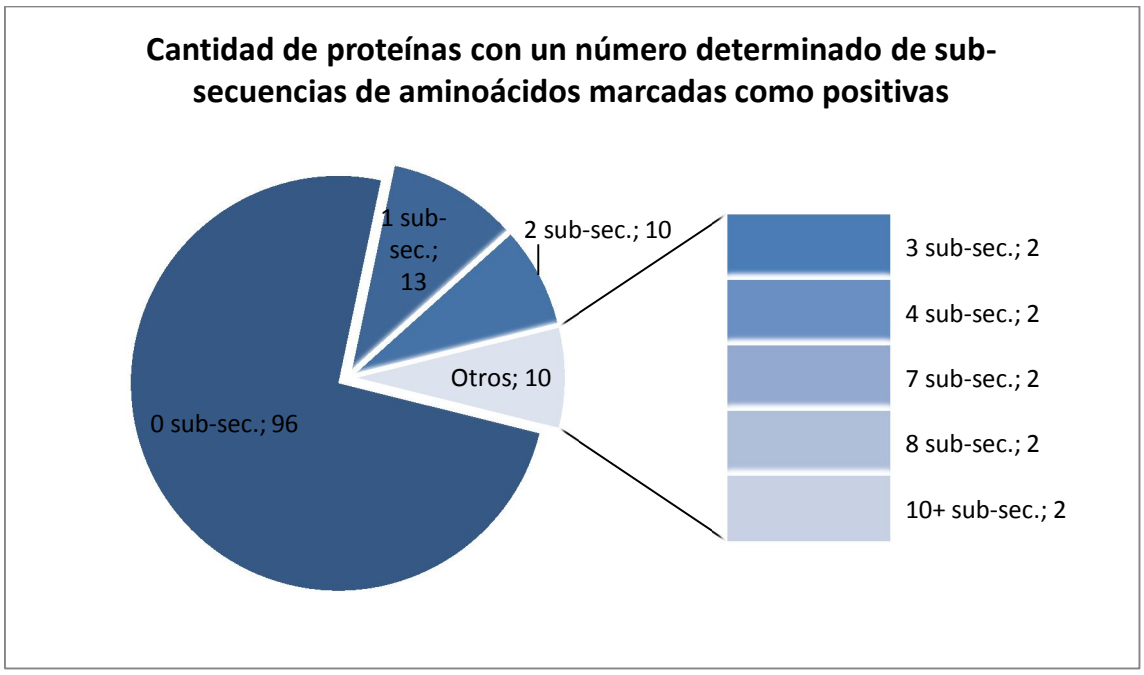
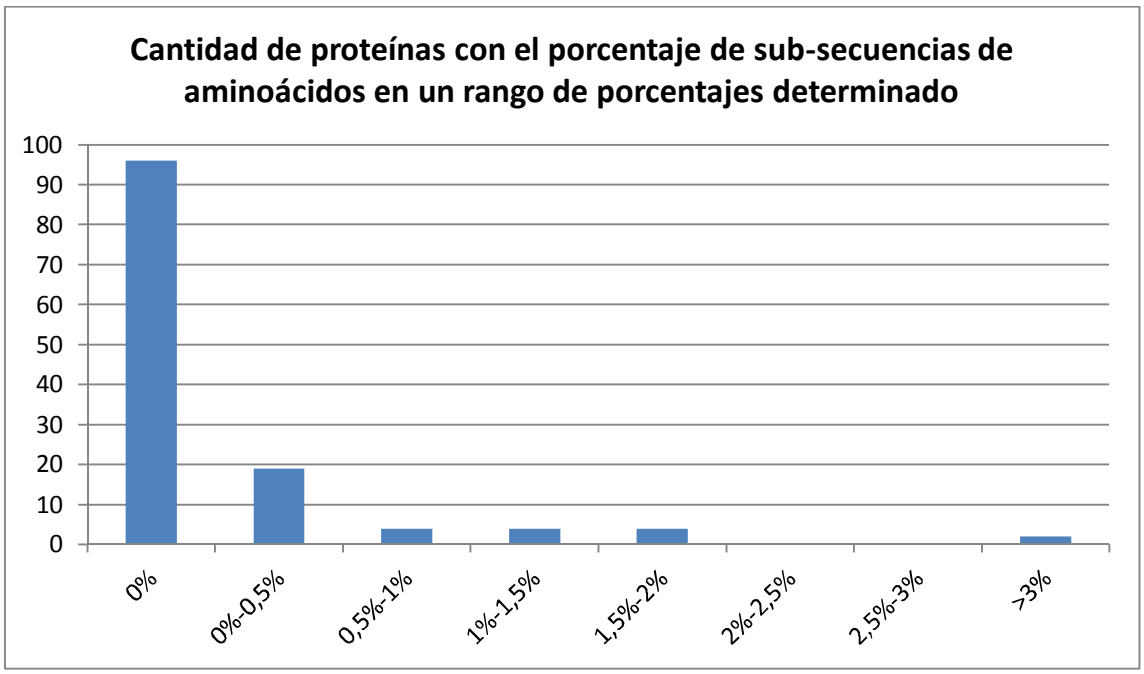


Figura 33: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo AdaBoostM1-Part



4.4.2.6. Modelo AdaBoostM1-J48

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 34: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo AdaBoostM1-J48

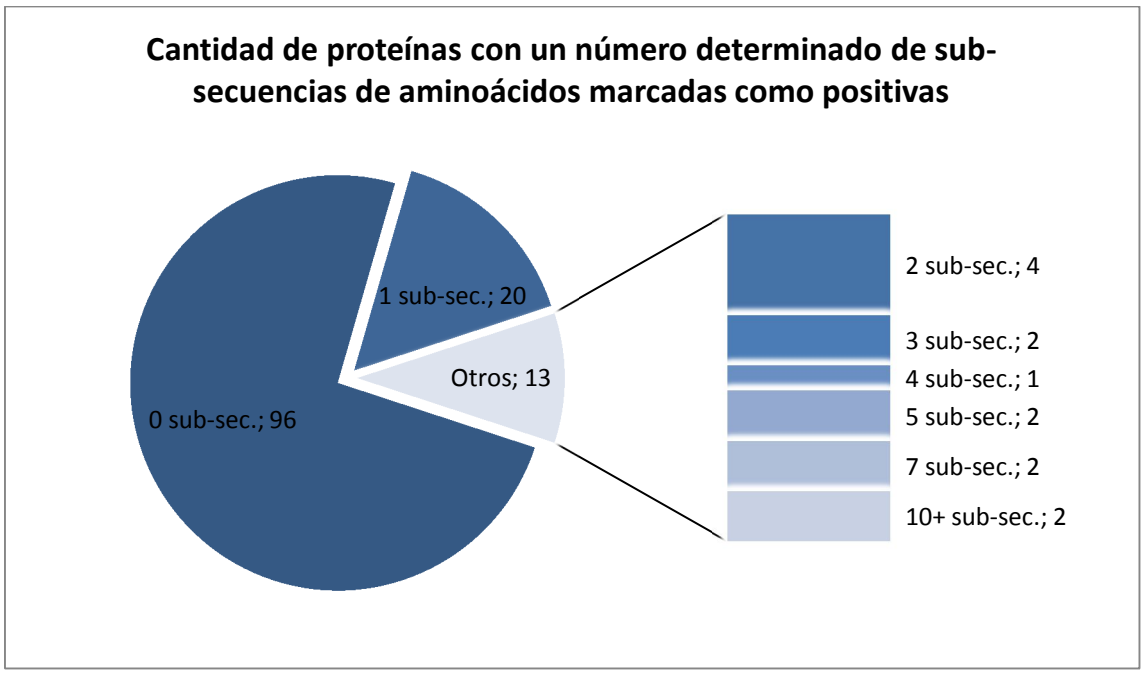
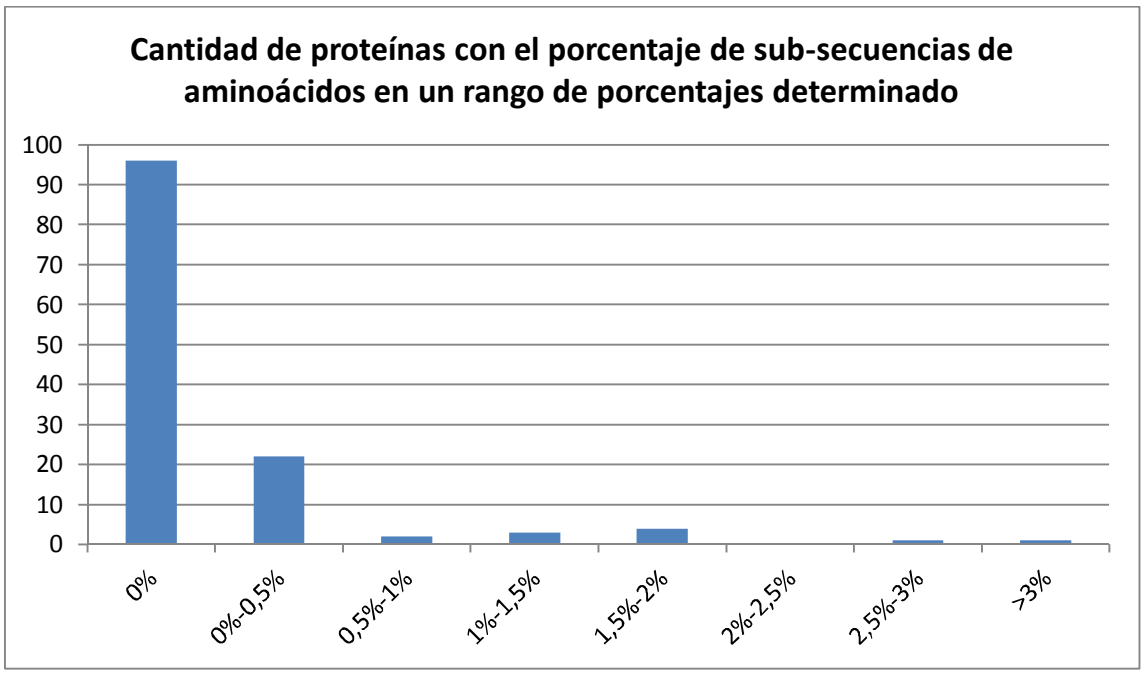


Figura 35: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo AdaBoostM1-J48



4.4.2.7. Modelo AdaBoostM1-NBTree

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 36: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo AdaBoostM1-NBTree

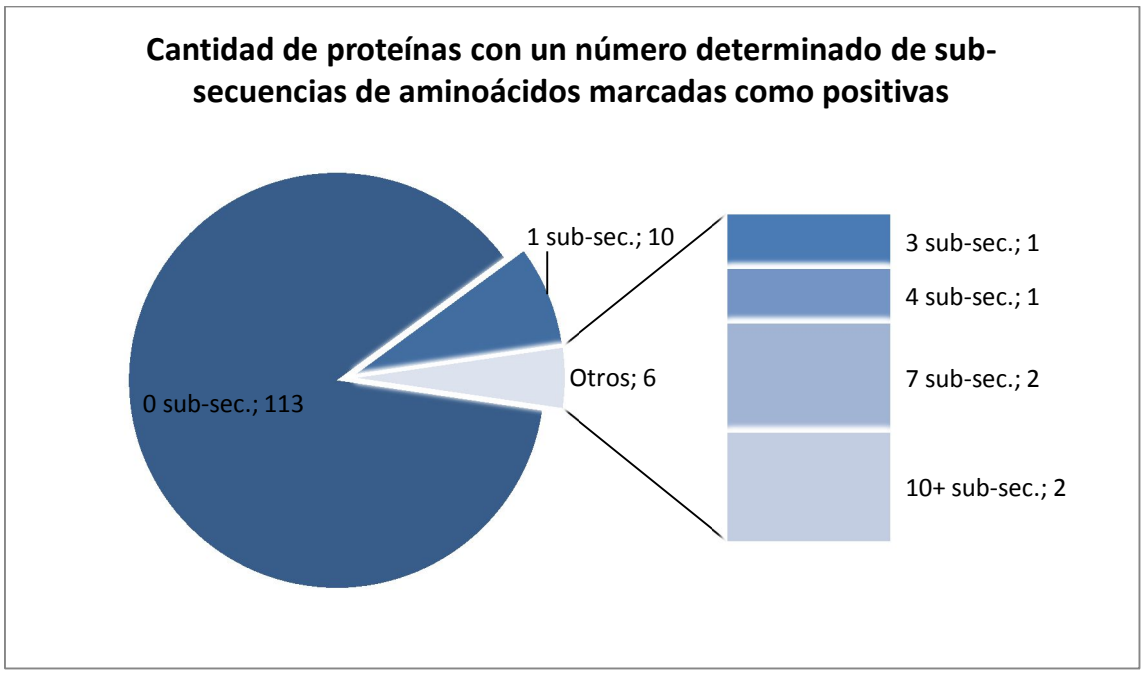
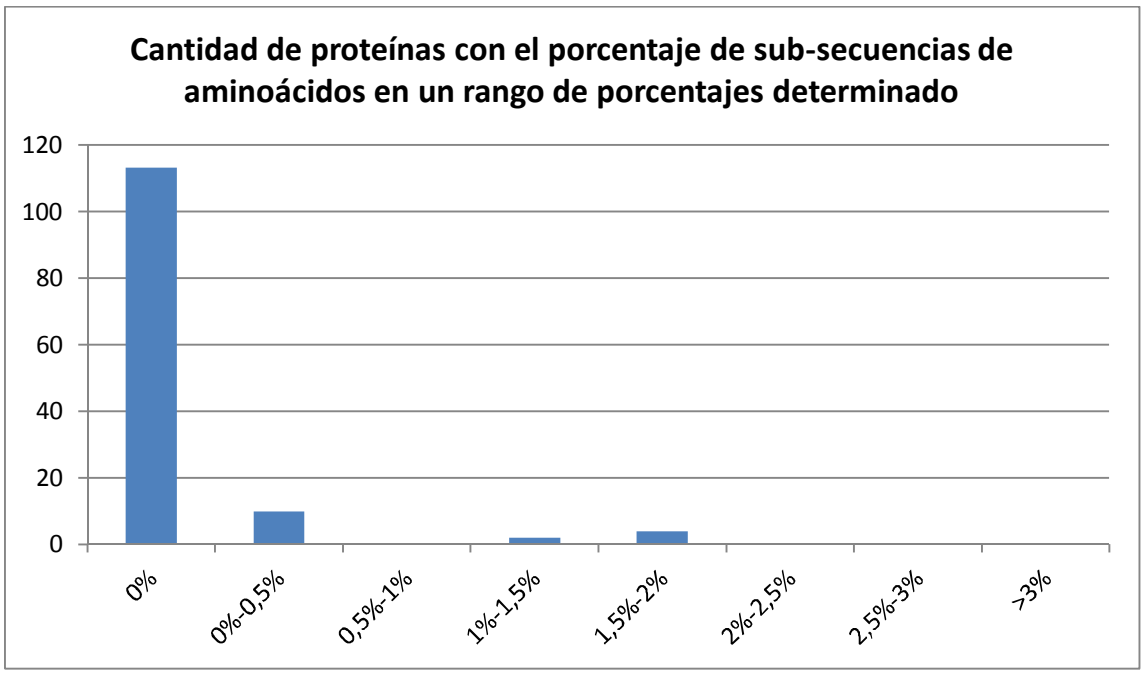


Figura 37: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo AdaBoostM1-NBTree



4.4.2.8. Modelo Bagging

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 38: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo Bagging

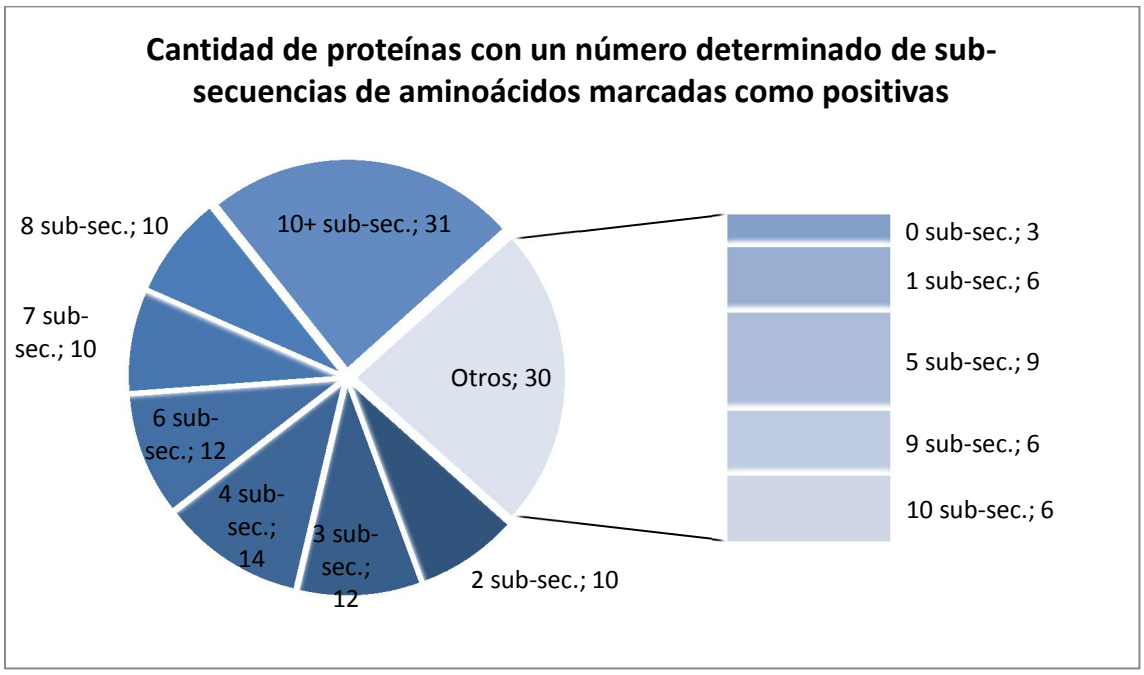
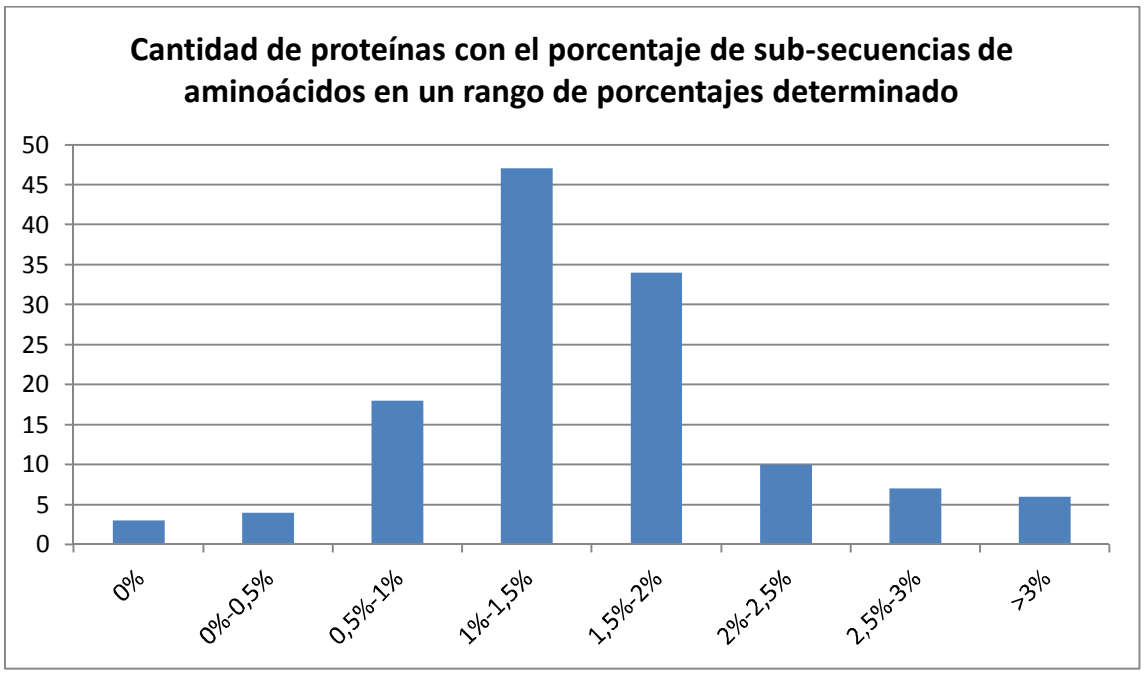


Figura 39: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo Bagging



4.4.2.9. Modelo Bagging-Part

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 40: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo Bagging-Part

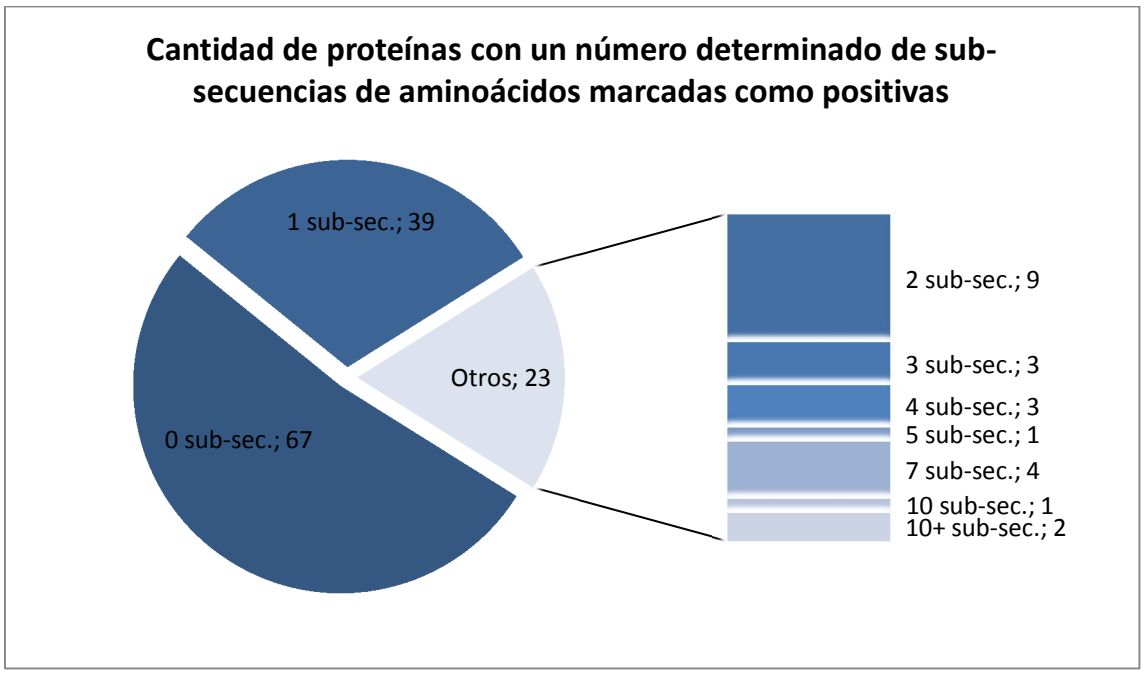
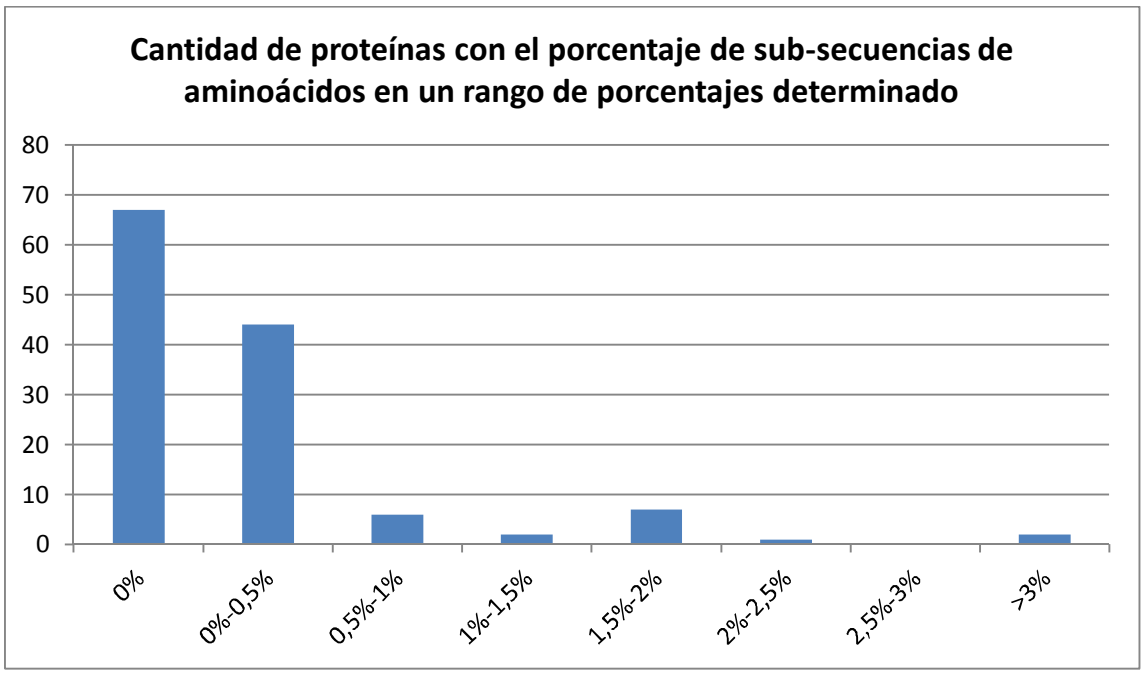


Figura 41: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo Bagging-Part



4.4.2.10. Modelo Bagging-J48

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 42: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo Bagging-J48

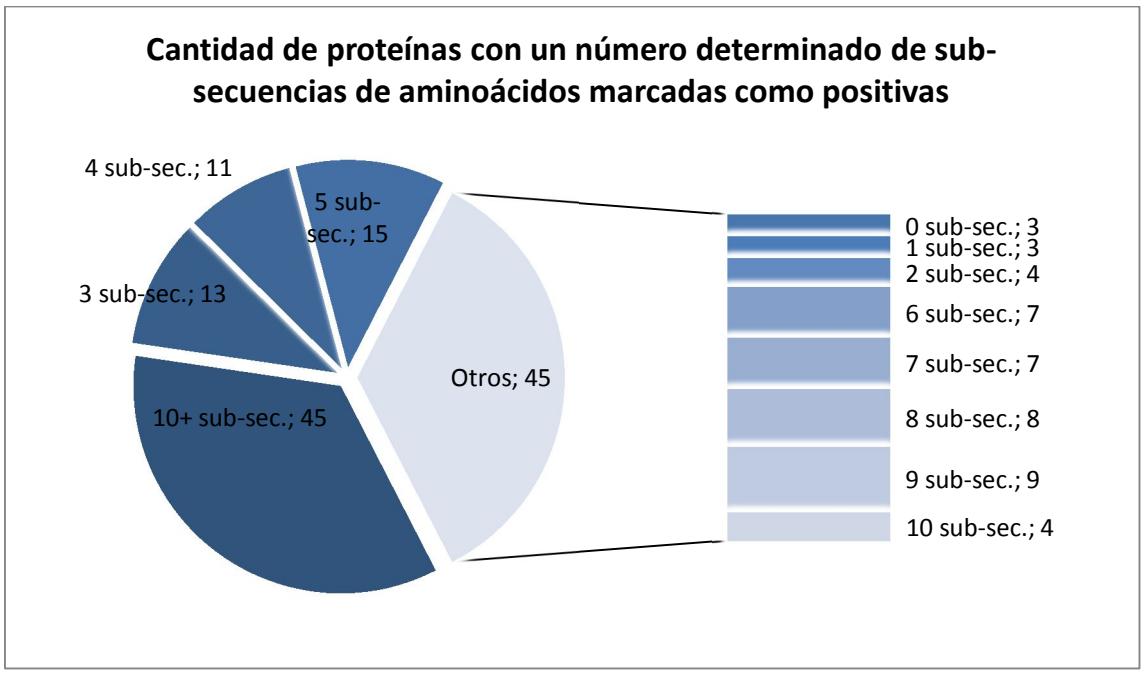
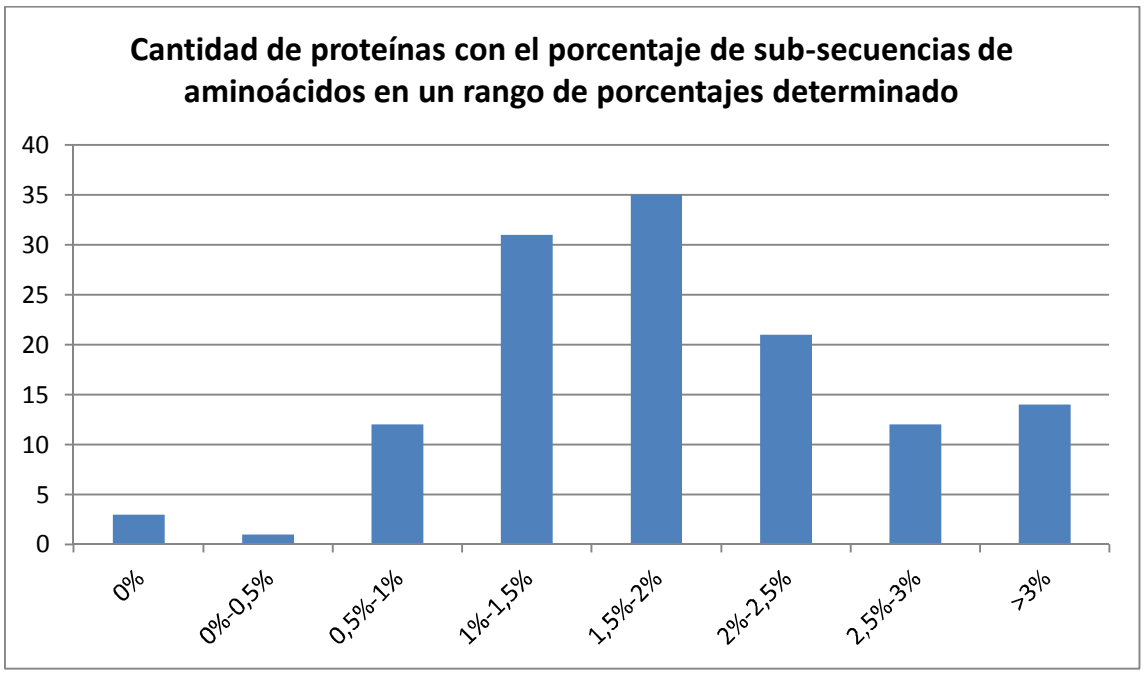


Figura 43: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo Bagging-J48



4.4.2.11. Modelo Bagging-NBTree

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 44: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo Bagging-NBTree

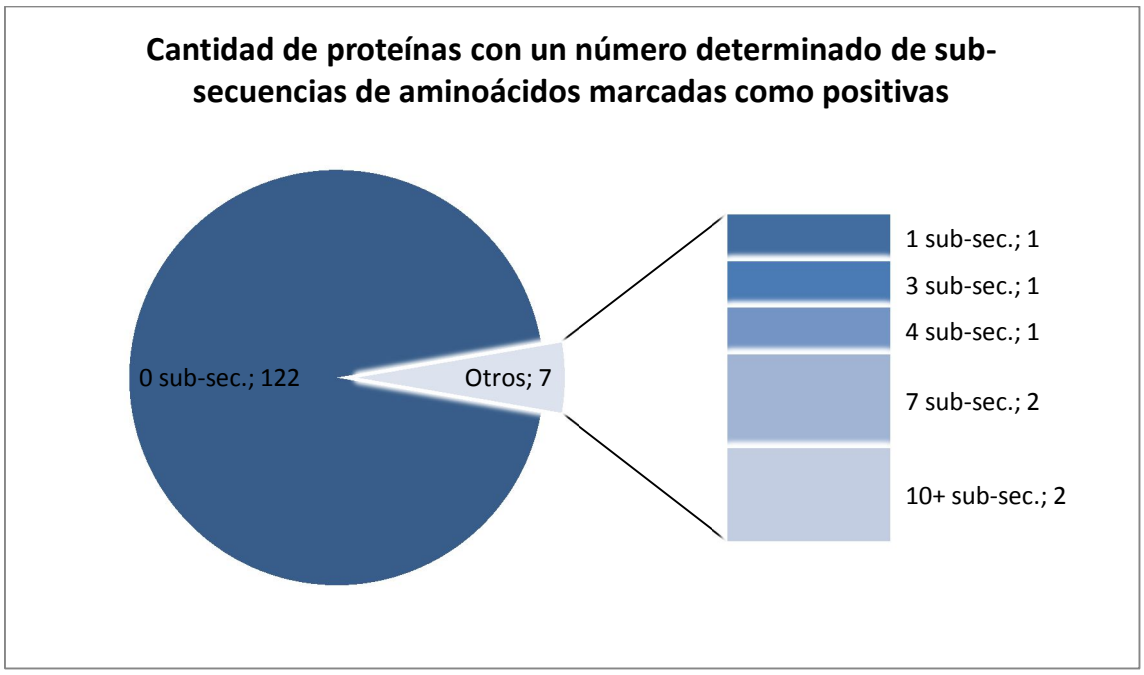
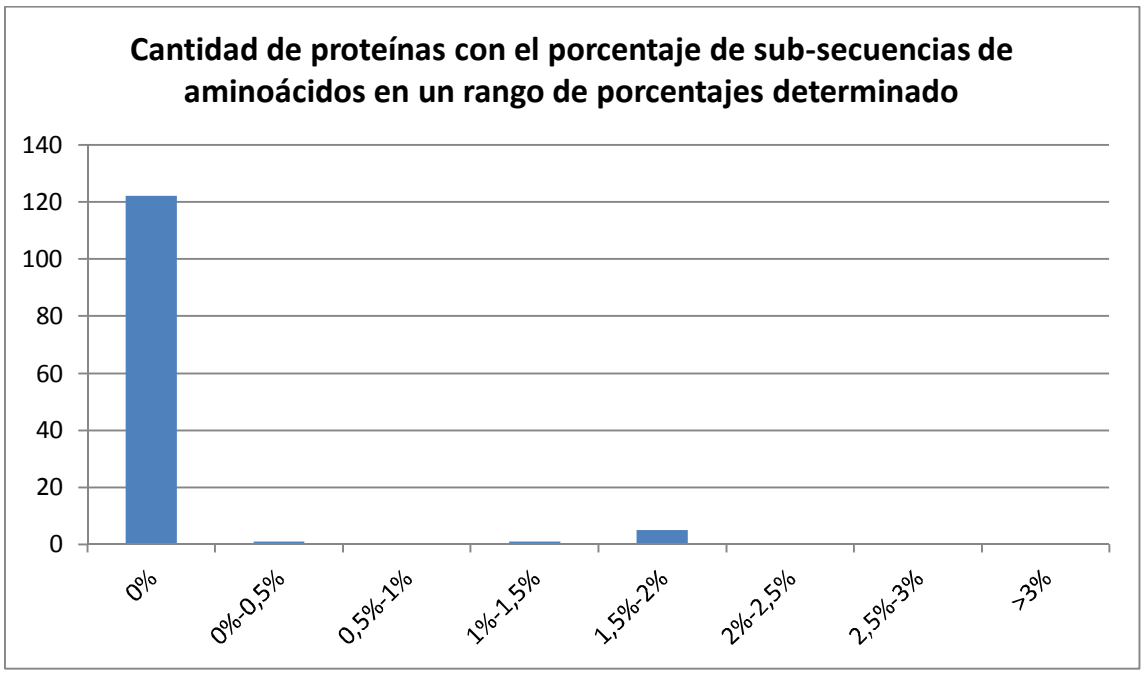


Figura 45: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo Bagging-NBTree



4.4.2.12. Modelo RacedIncrementalLogitBoost

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 46: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo RacedIncrementalLogitBoost

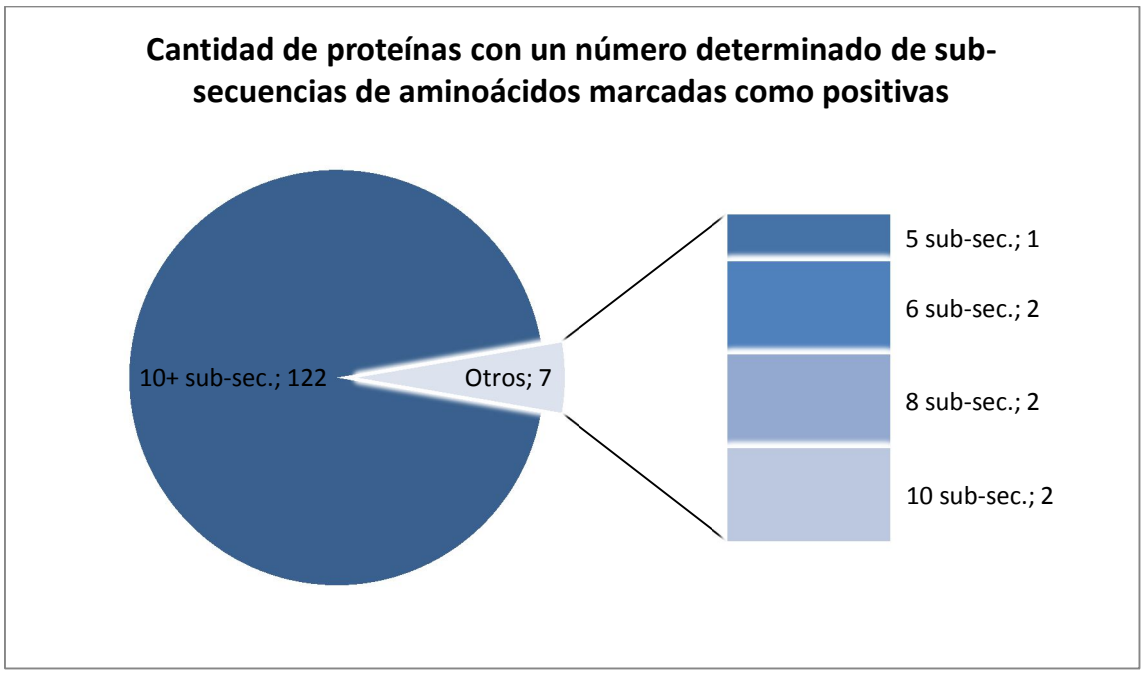
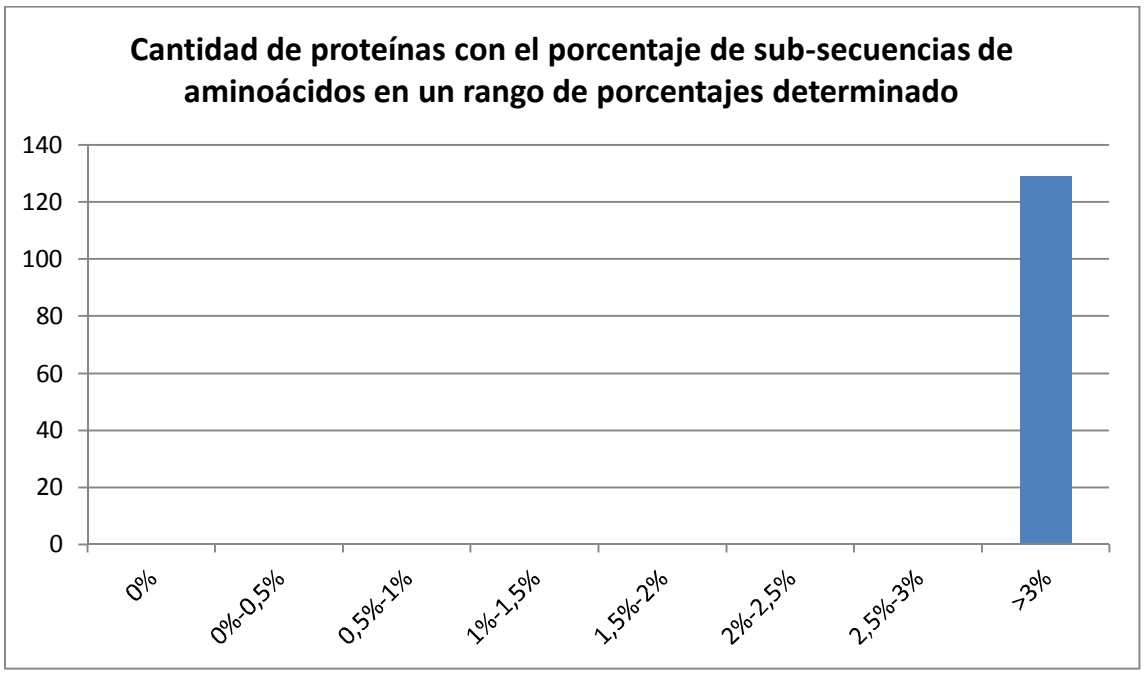


Figura 47: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo RacedIncrementalLogitBoost



4.4.2.13. Modelo RandomCommittee

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 48: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo RandomCommittee

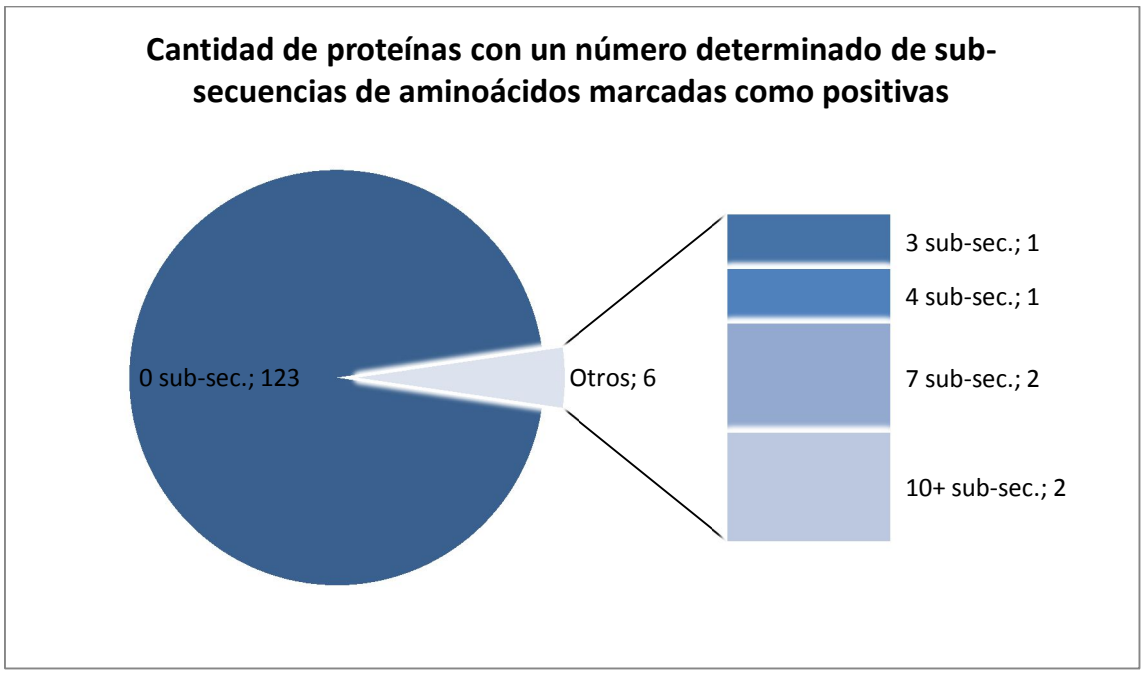
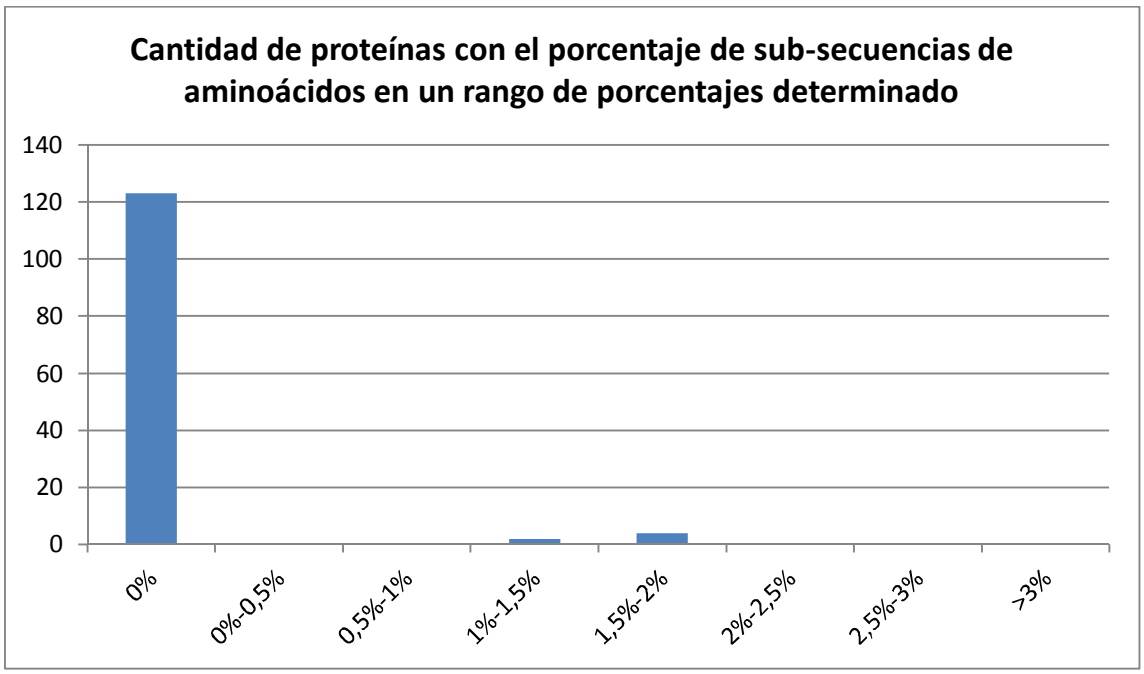


Figura 49: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo RandomCommittee



4.4.2.14. Modelo RotationForest-Part

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 50: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo RotationForest-Part

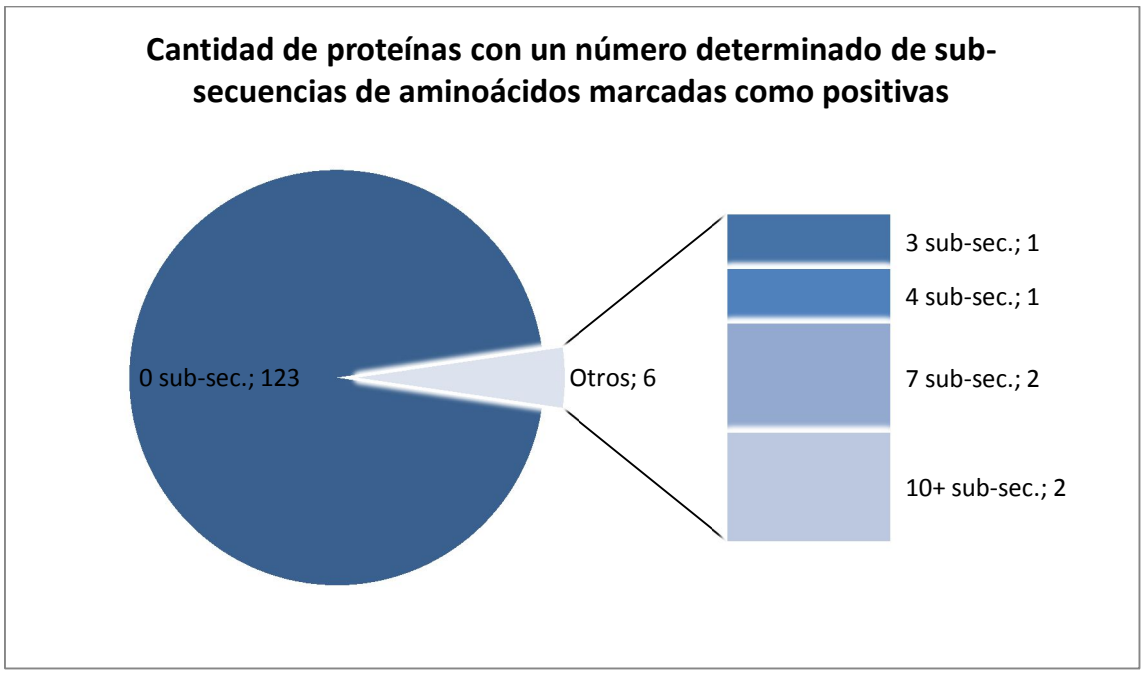
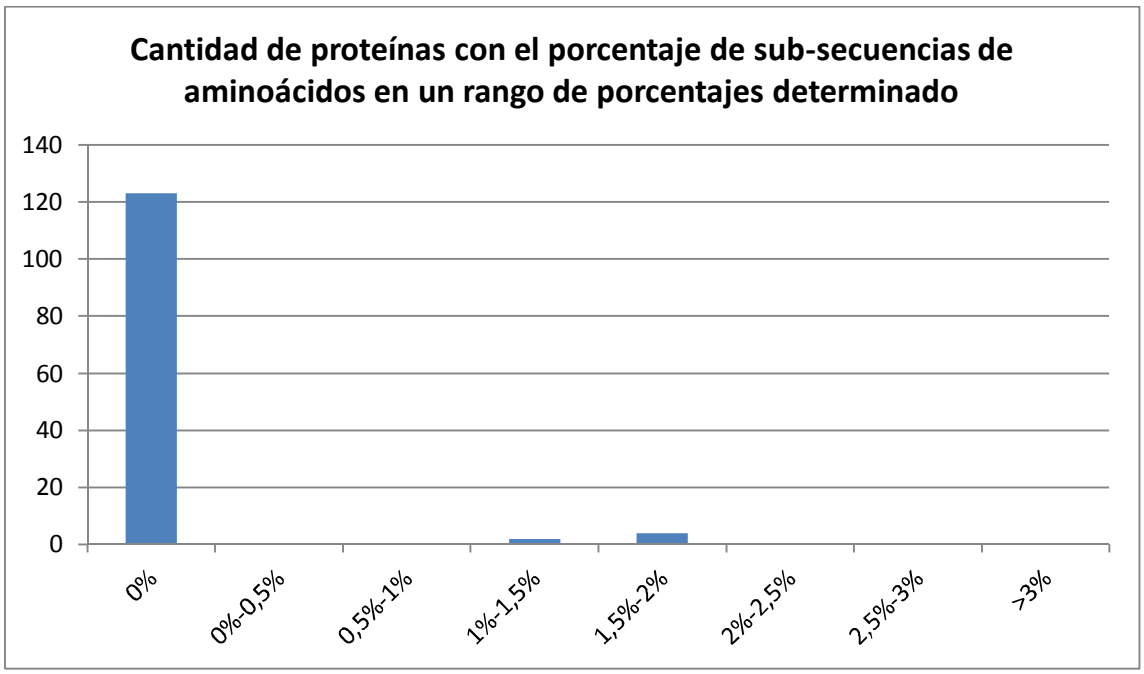


Figura 51: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo RotationForest-Part



4.4.2.15. Modelo RotationForest-J48

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas positivas este modelo:

Figura 52: Gráfica con el número de proteínas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo RotationForest-J48

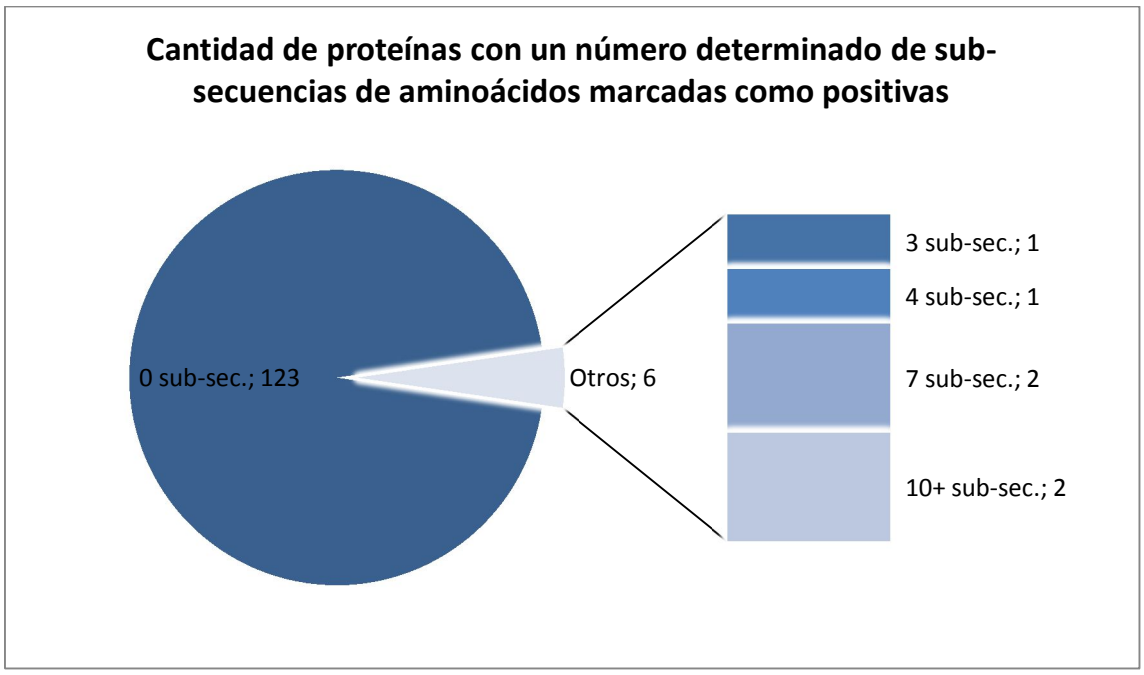
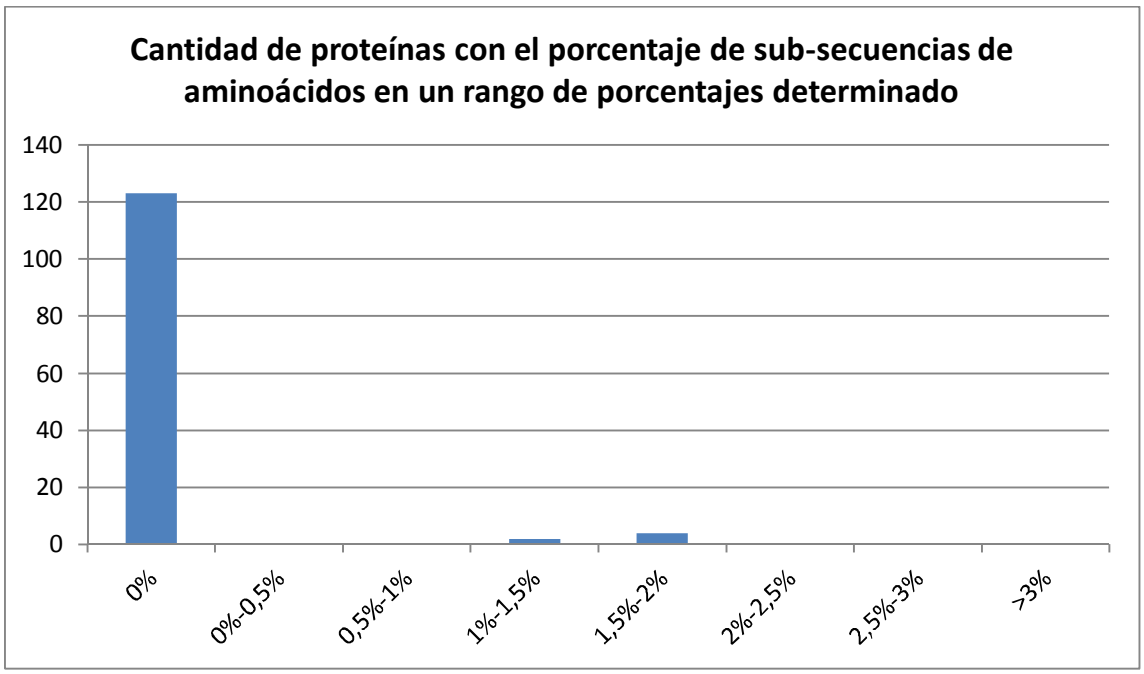


Figura 53: Gráfica que muestra la cantidad de proteínas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado para el modelo RotationForest-J48



4.4.3. Aplicación de los modelos generados sobre el fichero de proteínas negativas

A continuación se va a poner un resumen de los datos que resultan de aplicar cada uno de los modelos anteriormente generados, sobre el fichero de proteínas negativas.

Por cada uno de los modelos se incluyen unos gráficos que pretenden mostrar visualmente la clasificación de las proteínas. Para entenderlo mejor, los gráficos son:

- Un gráfico que indica cuántas proteínas hay con un número determinado de sub-secuencias de aminoácidos clasificadas como positivas.

- Un gráfico que indica el número de proteínas que hay en unos rangos de porcentaje calculados como relación de sub-secuencias de aminoácidos positivas sobre el total de sub-secuencias de la proteína. O dicho de otro modo, se refiere a un gráfico en el que se cuántas proteínas tienen en un X% y un Y% de sub-secuencias clasificadas como positivas.

4.4.3.1. Modelo Part

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 54: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo Part

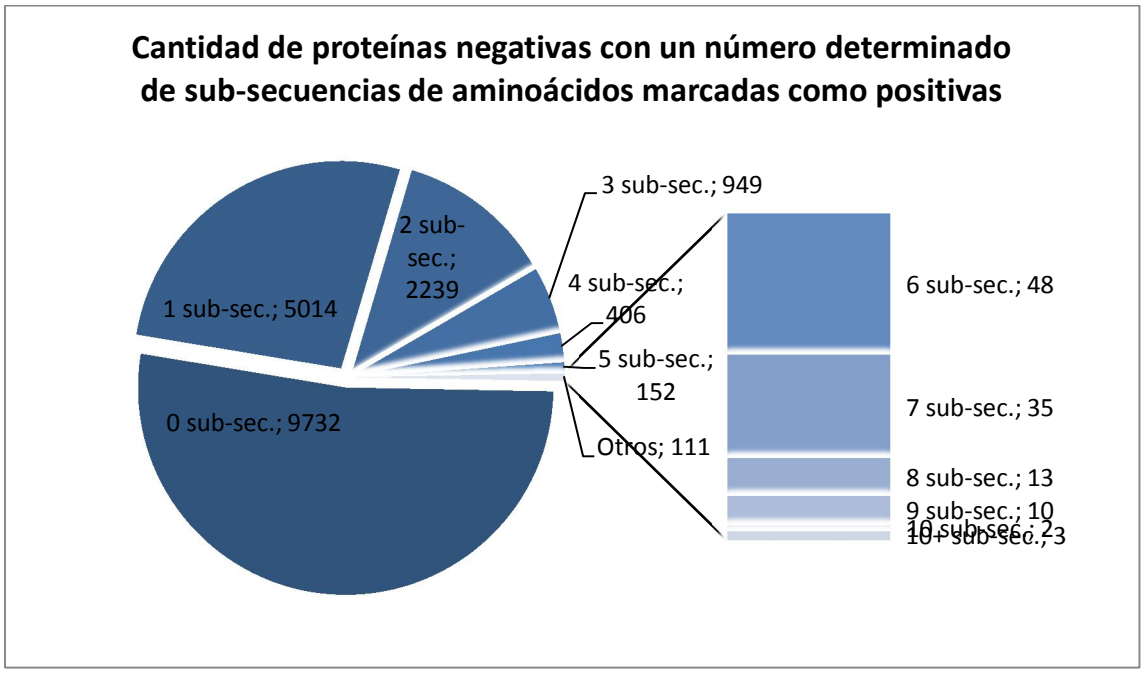
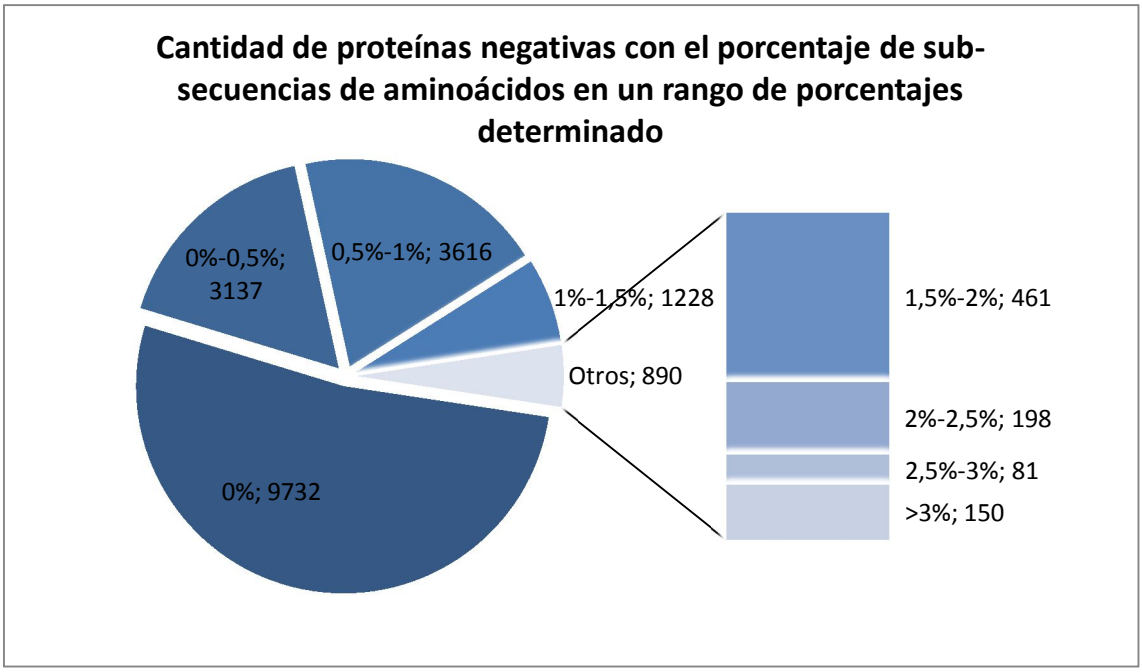


Figura 55: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo Part



Aunque este modelo parece que clasifica como positivas muchas sub-secuencias de las proteínas y, en principio, esto no debería ser así al tratarse de proteínas que inicialmente están catalogadas como negativas, las proteínas que más sub-secuencias tomadas como positivas, tanto en número como en porcentaje, tienen:

Tabla 18: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo Part

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
Número total de sub-secuencias clasificadas positivas[8]			
1cza:N	10	879	1,14%
2waq:B	12	1093	1,10%
2vz8:B	15	2474	0,61%
3cmu:A	12	2012	0,60%
3cmw:A	10	1668	0,60%
Mayor porcentaje de sub-secuencias clasificadas positivas sobre el total de sub-secuencias[9]			
1f2h:A	4	131	3,05%
2g1u:A	4	117	3,42%
2i8b:B	4	114	3,51%
2iih:A	4	119	3,36%
3by5:A	4	117	3,42%

[8] Con un mínimo de 10 sub-secuencias catalogadas como positivas.
[9] Criterio usado: un mínimo de 4 sub-secuencias catalogadas como positivas y un porcentaje sobre el total mayor al 3%.

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 19: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo Part

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	1	108	0,93%
1ozn:A	3	247	1,21%
1r4x:A	0	237	0,00%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	738	0,00%
2dba:A	0	110	0,00%
2e9g:A	1	93	1,08%
2iv9:A	0	200	0,00%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	2	397	0,50%
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	1	104	0,96%
2z5k:A	4	852	0,47%
2zej:B	1	146	0,68%
3ifq:C	1	69	1,45%
3lqv:P	0	1	0,00%

4.4.3.2. Modelo J48

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 56: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo J48

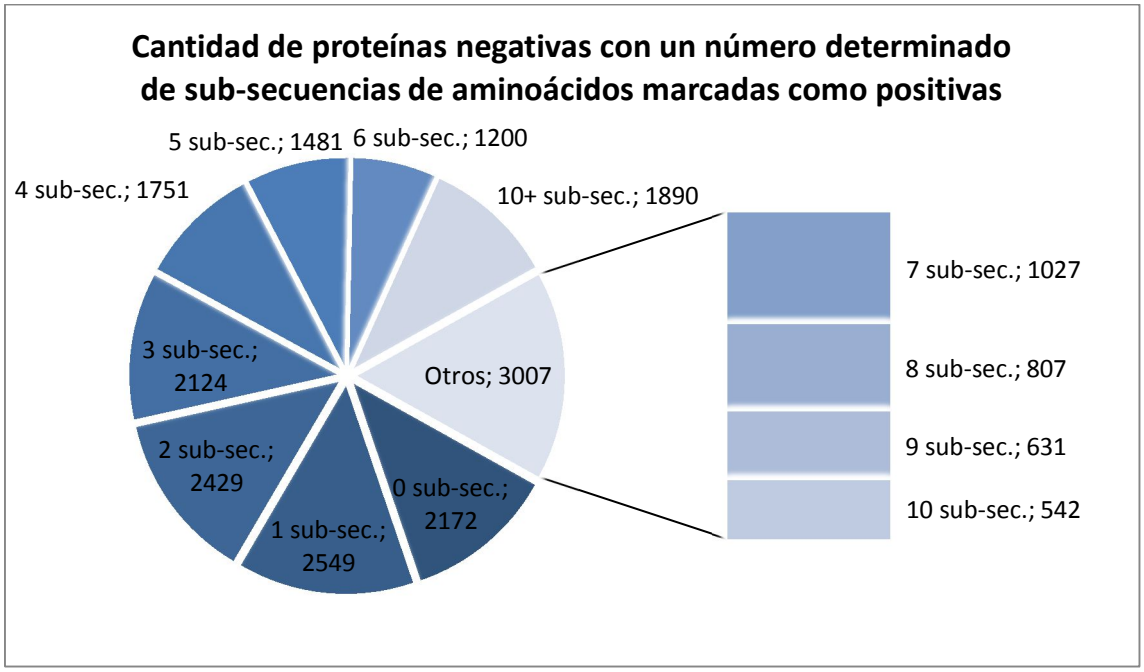
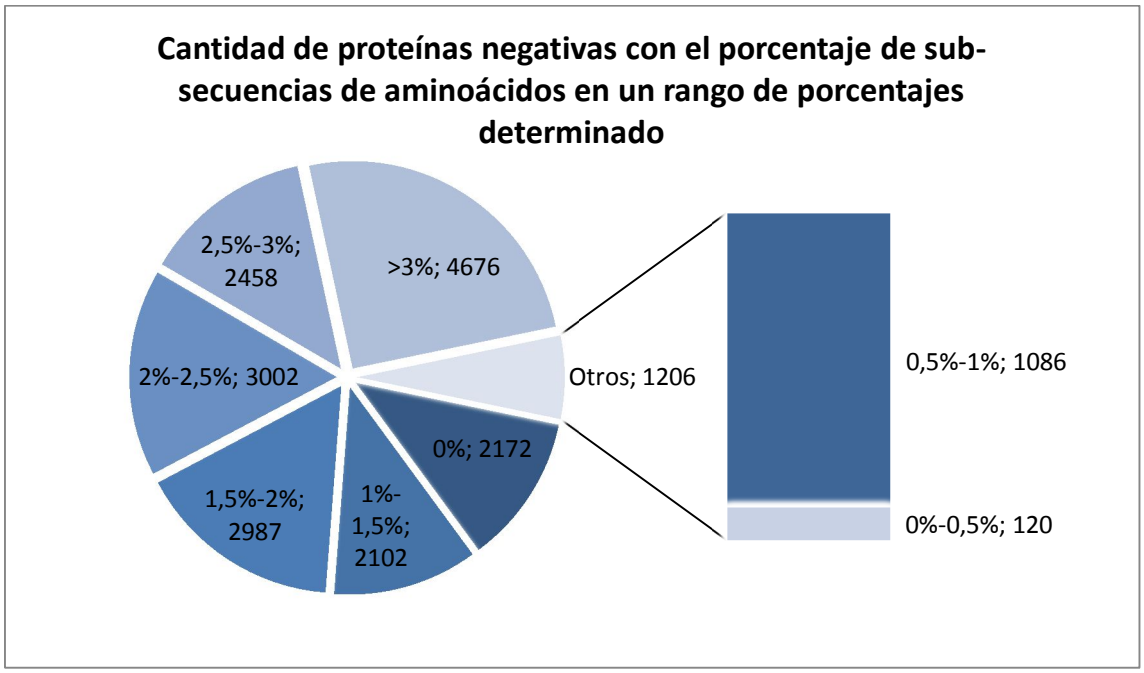


Figura 57: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo J48



Debido al reparto tan homogéneo que produce este modelo no destacan algunas proteínas sobre el resto. No obstante, las proteínas que más podrían destacar:

Tabla 20: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
Número total de sub-secuencias clasificadas positivas[10]			
2gsx:A	50	913	5,48%
2q7z:A	66	1893	3,49%
2vz8:B	63	2474	2,55%
3ecq:B	45	1493	3,01%
Mayor porcentaje de sub-secuencias clasificadas positivas sobre el total de sub-secuencias[11]			
1bcp:F	7	61	11,48%
1h8g:A	6	57	10,53%
1opi:A	7	66	10,61%
1zza:A	7	52	13,46%
2ge7:B	8	70	11,43%

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 21: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	1	108	0,93%
1ozn:A	10	247	4,05%
1r4x:A	8	237	3,38%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	16	738	2,17%
2dba:A	2	110	1,82%
2e9g:A	3	93	3,23%
2iv9:A	5	200	2,50%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado	13	397	3,27%

[10] Con un mínimo de 40 sub-secuencias catalogadas como positivas.
[11] Criterio usado: un mínimo de 6 sub-secuencias catalogadas como positivas y un porcentaje sobre el total mayor al 10%.

por incluir esta)			
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	3	104	2,88%
2z5k:A	32	852	3,76%
2zej:B	5	146	3,42%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

4.4.3.3. Modelo NBTree

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 58: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo NBTree

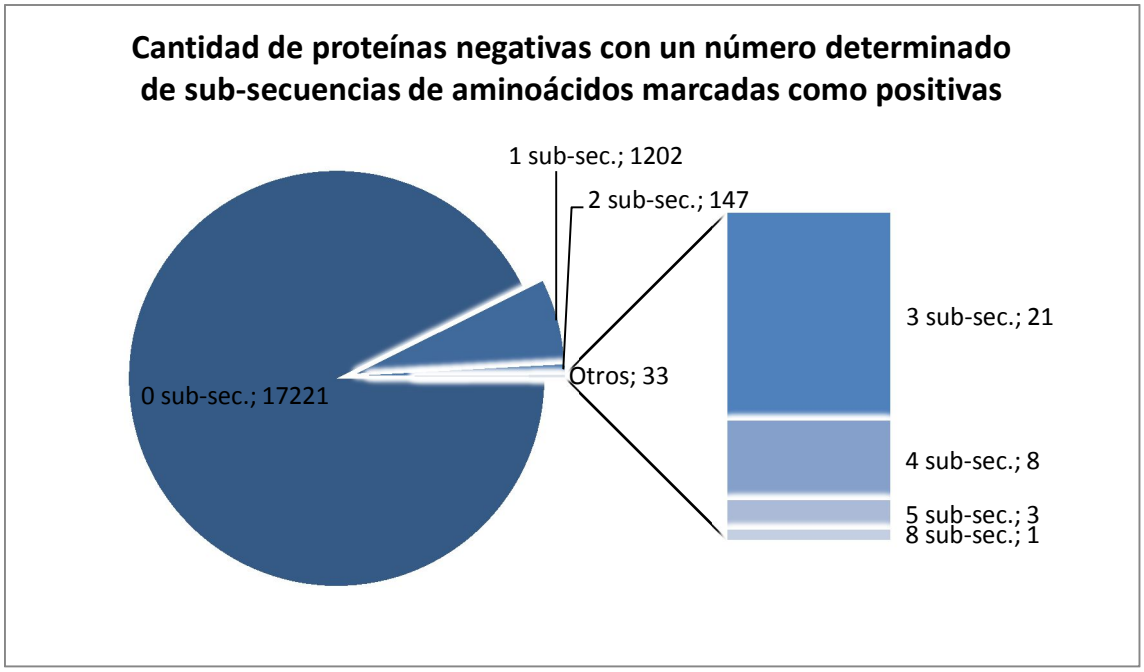
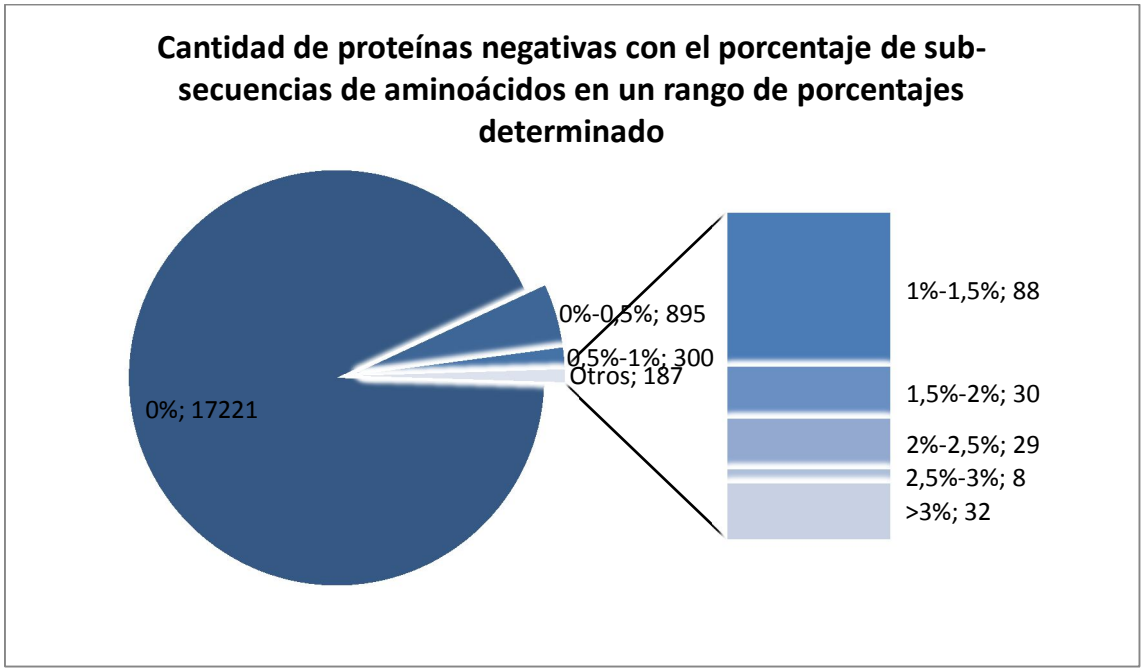


Figura 59: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo NBTree



Este modelo parece más restrictivo a la hora de clasificar como positivas las sub-secuencias de las proteínas, habiendo pocas, 33, proteínas con 3 o más sub-secuencias marcadas como positivas. Del total de resultados las que más destacan son:

Tabla 22: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo NBTree

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
Número total de sub-secuencias clasificadas positivas[12]			
2uvo:F	5	133	3,76%
2x2h:D	5	989	0,51%
3gau:A	5	1175	0,43%
1dx1:A	8	181	4,42%
Mayor porcentaje de sub-secuencias clasificadas positivas sobre el total de sub-secuencias[13]			
1dx1:A	8	181	4,42%
2kkg:A	4	47	8,51%
2uvo:F	5	133	3,76%
2w80:A	3	85	3,53%

[12] Con un mínimo de 5 sub-secuencias catalogadas como positivas.
[13] Criterio usado: un mínimo de 3 sub-secuencias catalogadas como positivas y un porcentaje sobre el total mayor al 3%.

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 23: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	0	108	0,00%
1ozn:A	0	247	0,00%
1r4x:A	0	237	0,00%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	738	0,00%
2dba:A	0	110	0,00%
2e9g:A	0	93	0,00%
2iv9:A	0	200	0,00%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	397	0,00%
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	104	0,00%
2z5k:A	0	852	0,00%
2zej:B	0	146	0,00%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

4.4.3.4. Modelo AdaBoostM1

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 60: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo AdaBoostM1

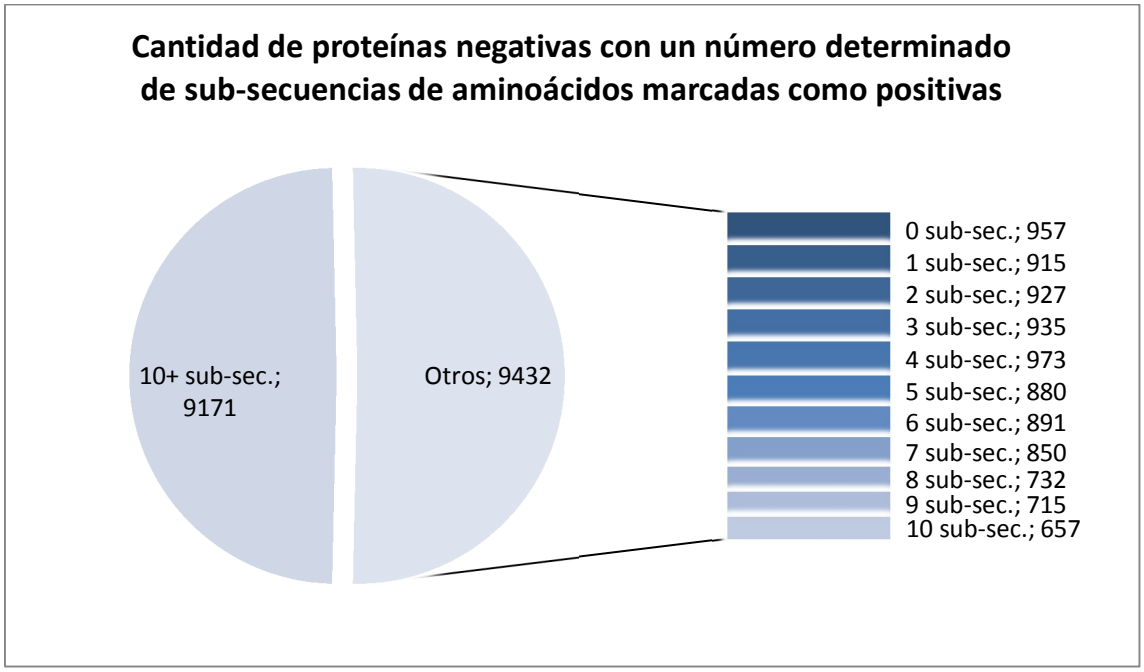
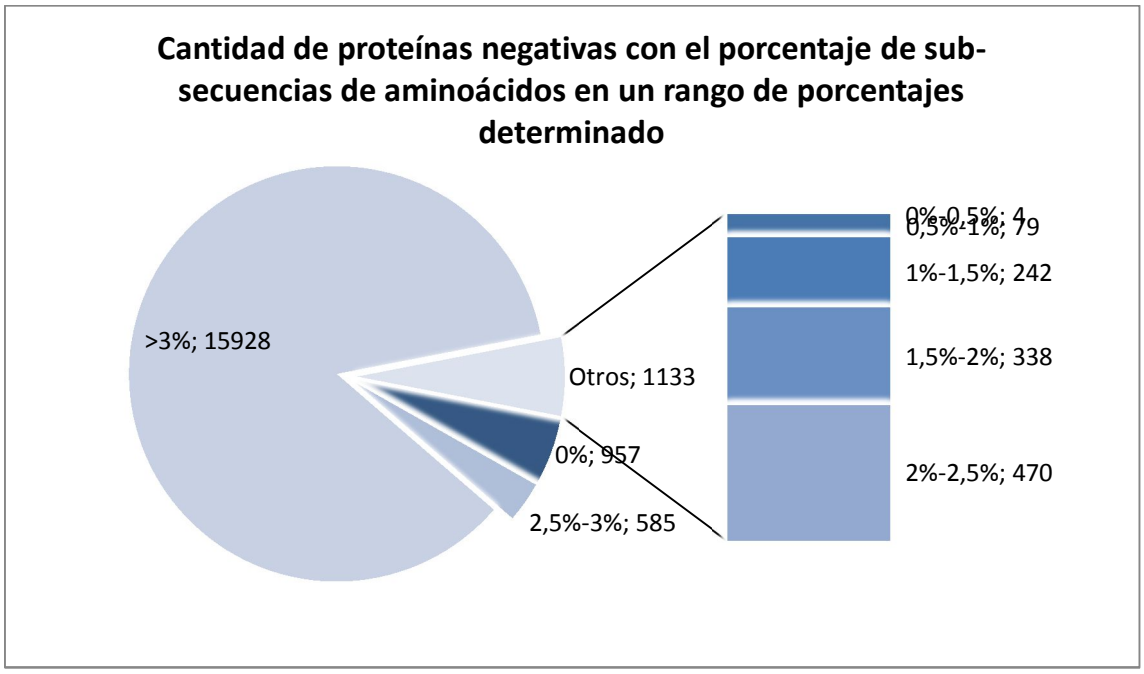


Figura 61: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo AdaBoostM1



Con este modelo se han clasificado muchas sub-secuencias como positivas. Baste como dato que aproximadamente la mitad de las proteínas inicialmente negativas han sido clasificadas con más de 10 sub-secuencias positivas. En cualquier caso, haciendo más estrictos los criterios de selección, las proteínas que más destacan con este modelo:

Tabla 24: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo AdaBoostM1

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
Número total de sub-secuencias clasificadas positivas[14]			
1w36:E	137	1142	12,00%
2eyq:A	136	1113	12,22%
2vz8:B	246	2474	9,94%
3cmu:A	141	2012	7,01%
3haz:A	148	963	15,37%
Mayor porcentaje de sub-secuencias clasificadas positivas sobre el total de sub-secuencias[15]			
2j9i:B	121	383	31,59%
2vpz:G	56	215	26,05%
3jyw:P	41	138	29,71%
3kaw:F	30	102	29,41%

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 25: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	4	108	3,70%
1ozn:A	43	247	17,41%
1r4x:A	9	237	3,80%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	43	738	5,83%
2dba:A	15	110	13,64%
2e9g:A	9	93	9,68%
2iv9:A	7	200	3,50%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína	17	397	4,28%

[14] Con un mínimo de 135 sub-secuencias catalogadas como positivas.
[15] Criterio usado: un mínimo de 25 sub-secuencias catalogadas como positivas y un porcentaje sobre el total mayor al 25%.

no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)			
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	3	104	2,88%
2z5k:A	70	852	8,22%
2zej:B	10	146	6,85%
3ifq:C	3	69	4,35%
3lqv:P	0	1	0,00%

4.4.3.5. Modelo AdaBoostM1-Part

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 62: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo AdaBoostM1-Part

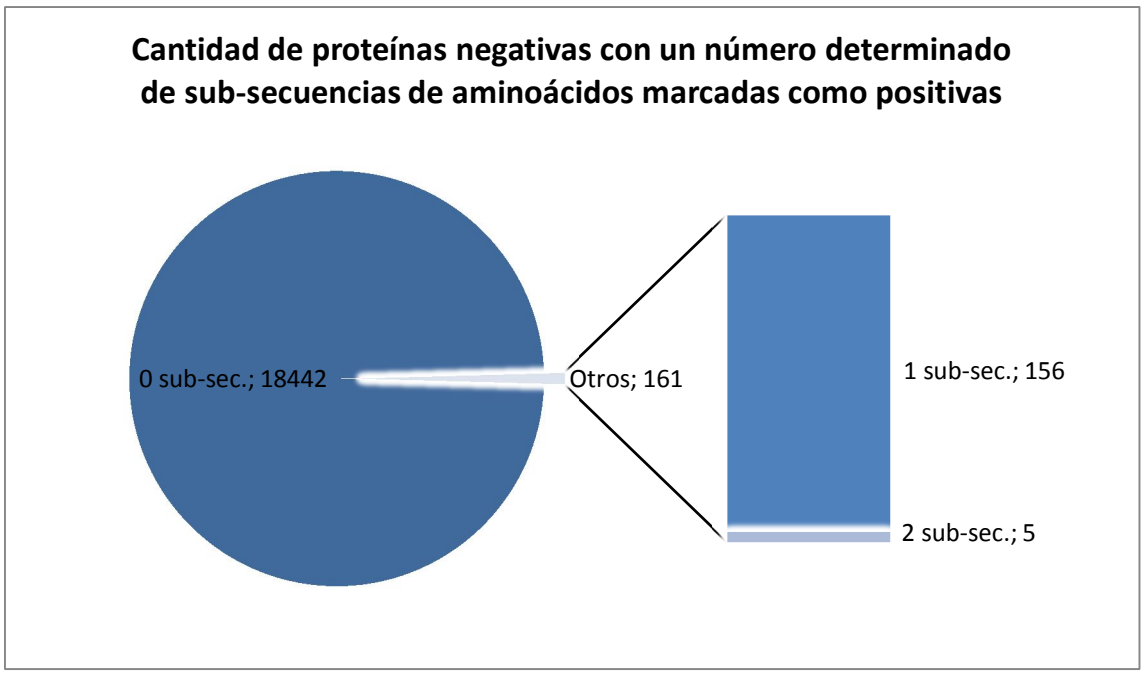
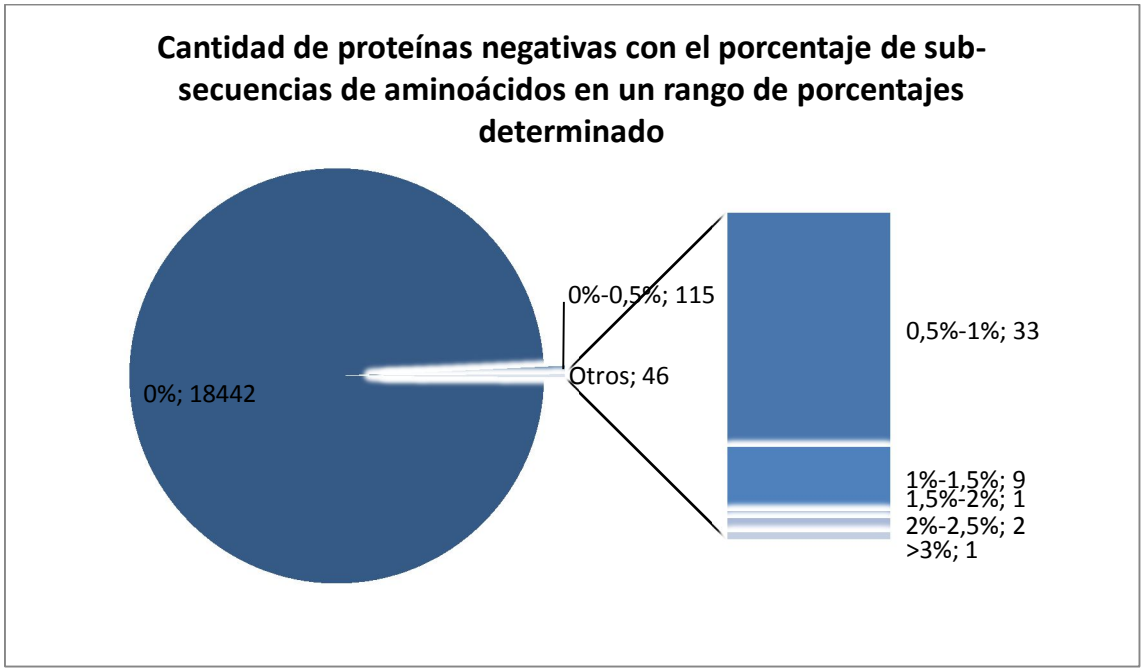


Figura 63: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo AdaBoostM1-Part



Este modelo es bastante restrictivo a la hora de clasificar como positivas las sub-secuencias de las proteínas, teniendo sólo 161 proteínas que tienen al menos 1 sub-secuencias marcada como positiva. Las que más destacan son:

Tabla 26: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo AdaBoostM1-Part

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
Número total de sub-secuencias clasificadas positivas[16]			
1t08:A	2	481	0,42%
2a3l:A	2	663	0,30%
2aja:B	2	338	0,59%
2qr4:A	2	549	0,36%
2z5k:A	2	852	0,23%
Mayor porcentaje de sub-secuencias clasificadas positivas sobre el total de sub-secuencias[17]			
1ghh:A	1	43	2,33%
2bf3:A	1	54	1,85%
2zkr:9	1	20	5,00%
3eqs:A	1	47	2,13%

[16] Con un mínimo de 5 sub-secuencias catalogadas como positivas.
[17] Criterio usado: un mínimo de 1 sub-secuencia catalogadas como positivas y un porcentaje sobre el total mayor al 1,5%.

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 27: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	0	108	0,00%
1ozn:A	0	247	0,00%
1r4x:A	0	237	0,00%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	738	0,00%
2dba:A	0	110	0,00%
2e9g:A	0	93	0,00%
2iv9:A	0	200	0,00%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	397	0,00%
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	104	0,00%
2z5k:A	2	852	0,23%
2zej:B	0	146	0,00%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

4.4.3.6. Modelo AdaBoostM1-J48

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 64: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo AdaBoostM1-J48

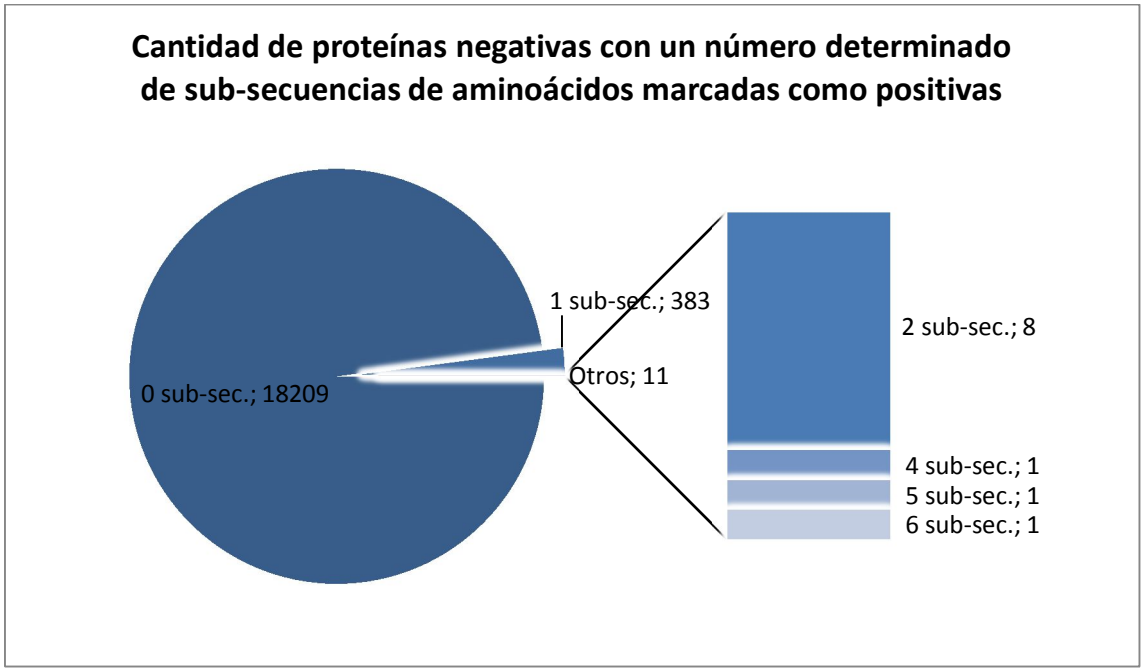
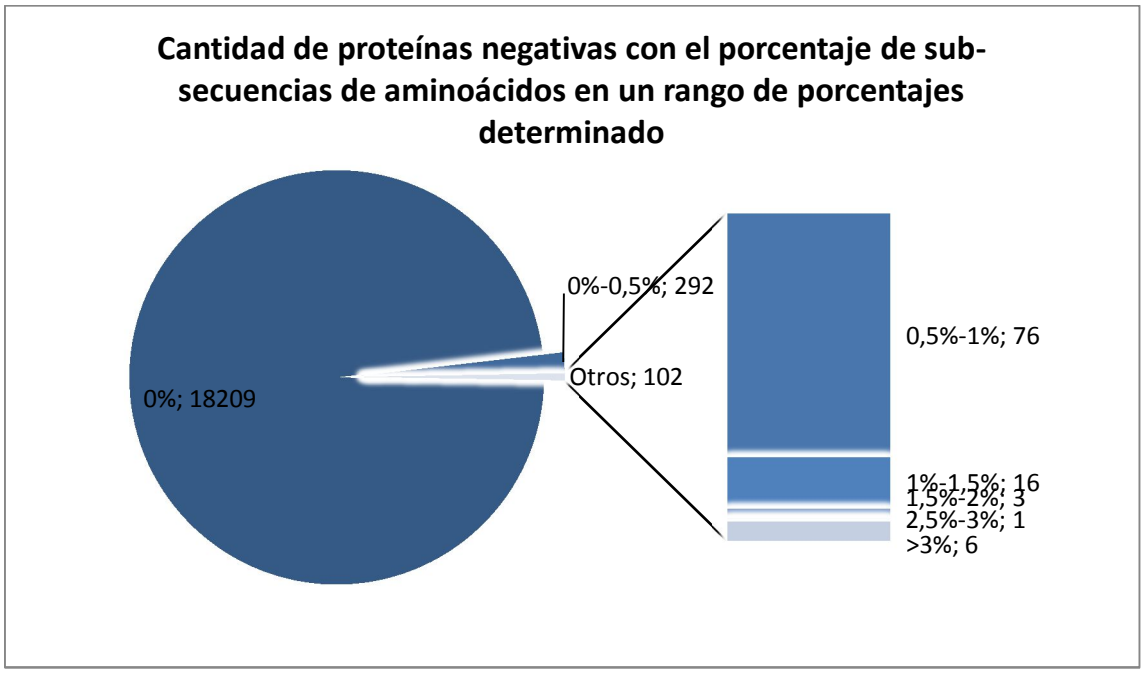


Figura 65: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo AdaBoostM1-J48



Este modelo es muy restrictivo a la hora de clasificar como positivas las sub-secuencias de las proteínas, ya que sólo 394 proteínas de las 18.603 tienen al menos 1 sub-secuencia marcada como positiva. Las que más destacan son:

Tabla 28: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo AdaBoostM1-J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
Número total de sub-secuencias clasificadas positivas[18]			
1aud:A	2	63	3,17%
1cx0:A	2	57	3,51%
1m1c:A	2	642	0,31%
1w36:C	2	1084	0,18%
1wa5:B	2	492	0,41%
2aja:B	2	338	0,59%
2bji:A	2	239	0,84%
3cmu:A	6	2012	0,30%
3cmv:A	4	1319	0,30%
3cmw:A	5	1668	0,30%
3pgw:P	2	244	0,82%
Mayor porcentaje de sub-secuencias clasificadas positivas sobre el total de sub-secuencias[19]			
1aud:A	2	63	3,17%
1b28:A	1	15	6,67%
1cx0:A	2	57	3,51%
1kg1:A	1	22	4,55%
1nla:A	1	26	3,85%
1y3k:A	1	39	2,56%
2ct2:A	1	50	2,00%
2jpc:A	1	23	4,35%

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 29: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	0	108	0,00%
1ozn:A	0	247	0,00%
1r4x:A	0	237	0,00%

[18] Con un mínimo de 2 sub-secuencias catalogadas como positivas.
[19] Criterio usado: un mínimo de 1 sub-secuencia catalogada como positiva y un porcentaje sobre el total mayor o igual al 2%.

1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	738	0,00%
2dba:A	0	110	0,00%
2e9g:A	0	93	0,00%
2iv9:A	0	200	0,00%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	397	0,00%
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	104	0,00%
2z5k:A	1	852	0,12%
2zej:B	0	146	0,00%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

4.4.3.7. Modelo AdaBoostM1-NBTree

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 66: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo AdaBoostM1-NBTree

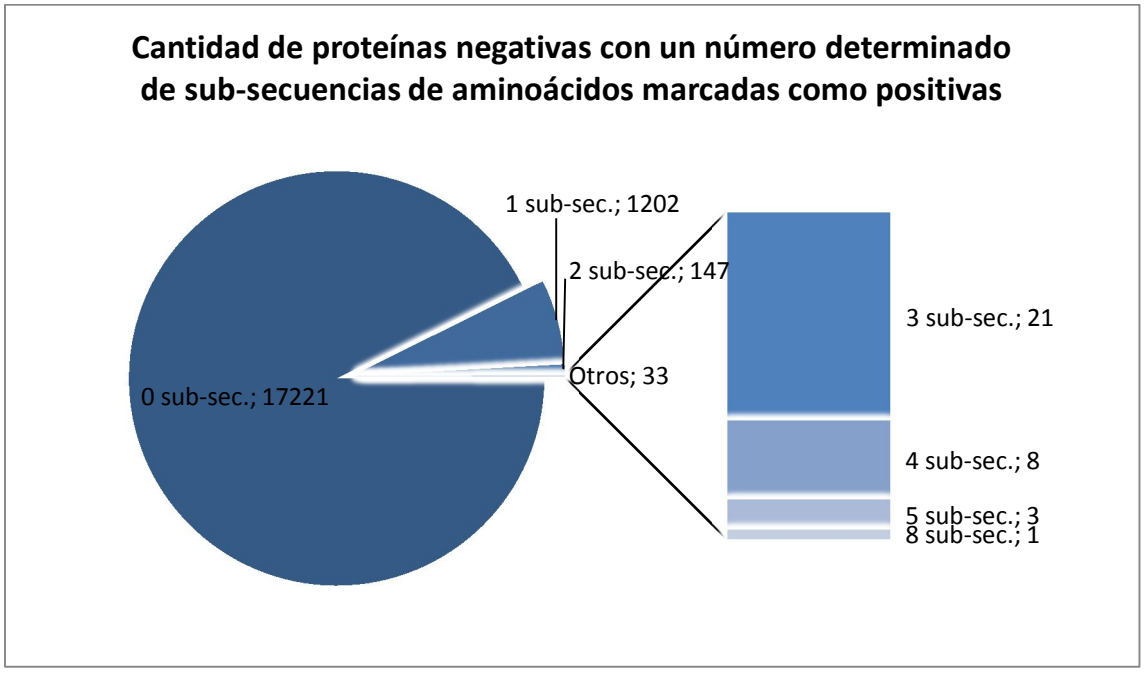
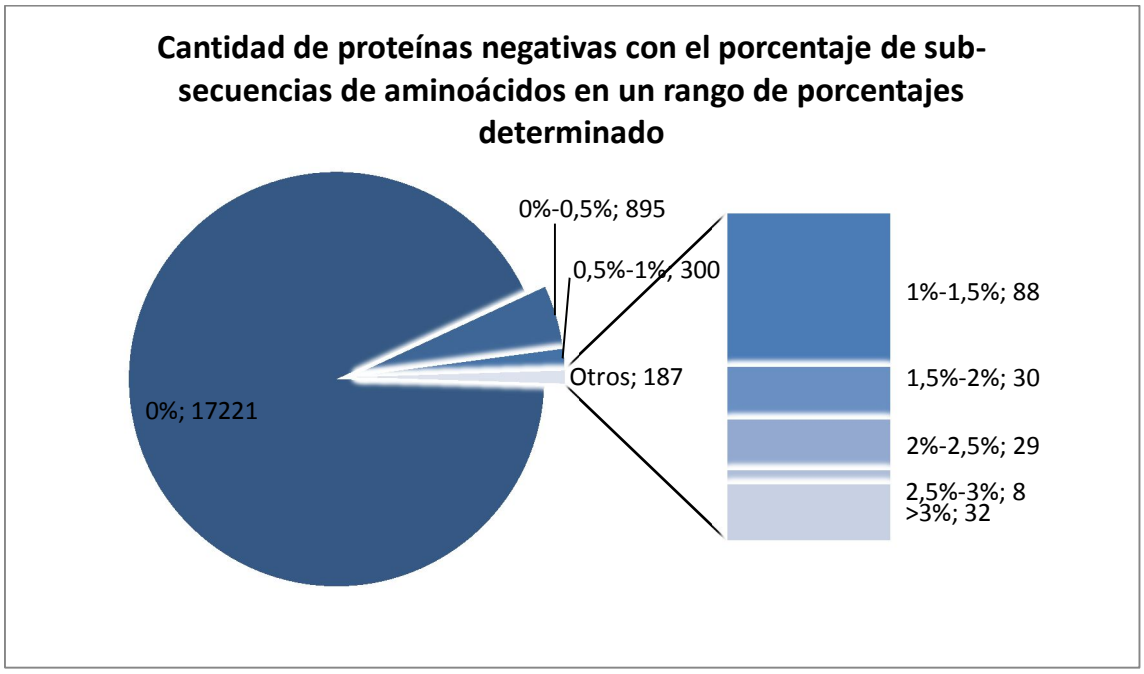


Figura 67: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo AdaBoostM1-NBTree



Este modelo es restrictivo a la hora de clasificar como positivas las sub-secuencias de las proteínas, y sólo 180 proteínas de las 18.603 tienen al menos 2 sub-secuencias marcada como positivas. Las que más destacan son:

Tabla 30: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo AdaBoostM1-NBTree

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
Número total de sub-secuencias clasificadas positivas[20]			
1dx1:A	8	181	4,42%
2uvo:F	5	133	3,76%
2x2h:D	5	989	0,51%
3gau:A	5	1175	0,43%
Mayor porcentaje de sub-secuencias clasificadas positivas sobre el total de sub-secuencias[21]			
1dx1:A	8	181	4,42%
2kkg:A	4	47	8,51%
2uvo:F	5	133	3,76%
2w80:A	3	85	3,53%

[20] Con un mínimo de 5 sub-secuencias catalogadas como positivas.
[21] Criterio usado: un mínimo de 3 sub-secuencias catalogadas como positivas y un porcentaje sobre el total mayor o igual al 3%.

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 31: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	0	108	0,00%
1ozn:A	0	247	0,00%
1r4x:A	0	237	0,00%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	738	0,00%
2dba:A	0	110	0,00%
2e9g:A	0	93	0,00%
2iv9:A	0	200	0,00%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	397	0,00%
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	104	0,00%
2z5k:A	0	852	0,00%
2zej:B	0	146	0,00%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

4.4.3.8. Modelo Bagging

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 68: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo Bagging

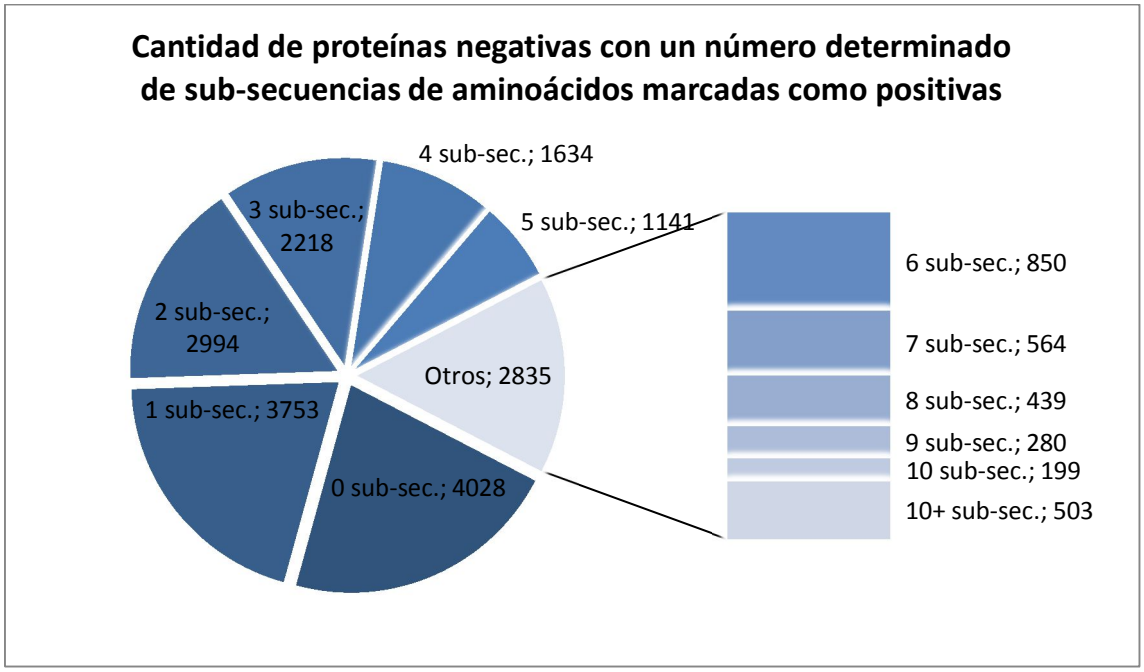
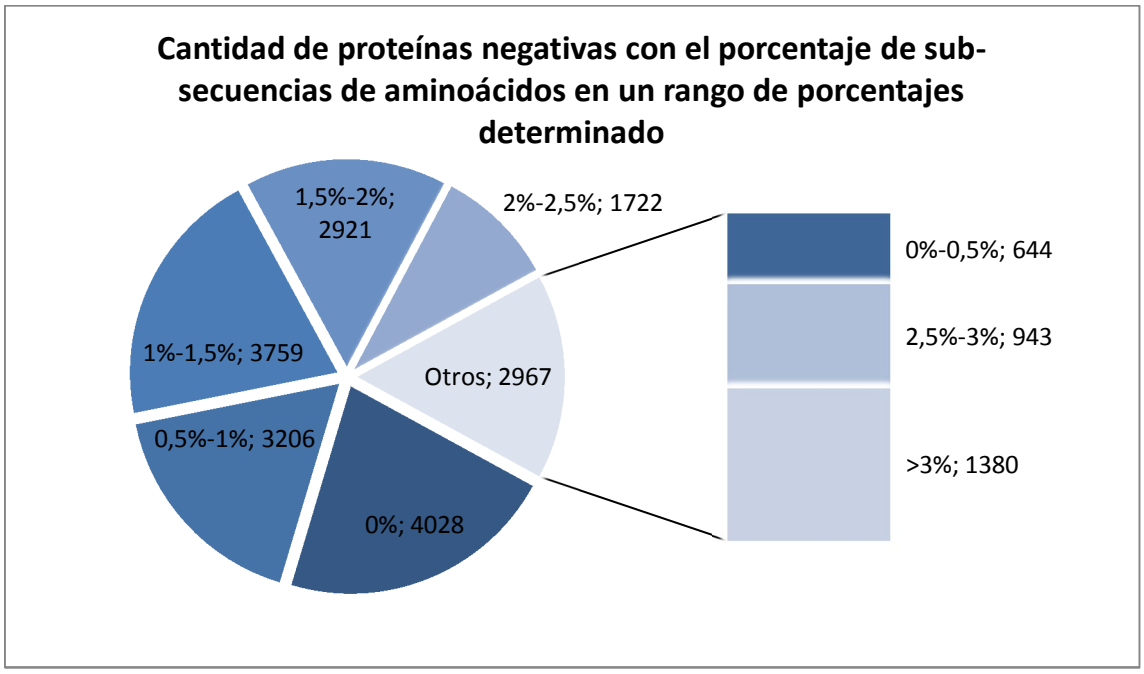


Figura 69: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo Bagging



Debido al reparto tan homogéneo que produce este modelo no destacan algunas proteínas sobre el resto. No obstante, las proteínas que más podrían destacar:

Tabla 32: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo Bagging

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
Número total de sub-secuencias clasificadas positivas[22]			
2gsx:A	27	913	2,96%
2q7z:A	74	1893	3,91%
2vz8:B	44	2474	1,78%
3cu7:A	28	1638	1,71%
3gau:A	27	1175	2,30%
Mayor porcentaje de sub-secuencias clasificadas positivas sobre el total de sub-secuencias[23]			
1hcc:A	3	21	14,29%
1koy:A	3	24	12,50%
1kve:C	3	25	12,00%
1m56:J	4	9	30,77%
1syx:B	5	48	10,42%
1vaz:A	5	50	10,00%
3o70:A	3	30	10,00%

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 33: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	2	108	1,85%
1ozn:A	7	247	2,83%
1r4x:A	7	237	2,95%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	16	738	2,17%
2dba:A	2	110	1,82%
2e9g:A	3	93	3,23%
2iv9:A	1	200	0,50%
2vgl:M (la proteína que destacaba en el	2	397	0,50%

[22] Con un mínimo de 27 sub-secuencias catalogadas como positivas.
[23] Criterio usado: un mínimo de 3 sub-secuencias catalogadas como positivas y un porcentaje sobre el total mayor o igual al 10%.

artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)			
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	1	104	0,96%
2z5k:A	15	852	1,76%
2zej:B	3	146	2,05%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

4.4.3.9. Modelo Bagging-Part

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 70: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo Bagging-Part

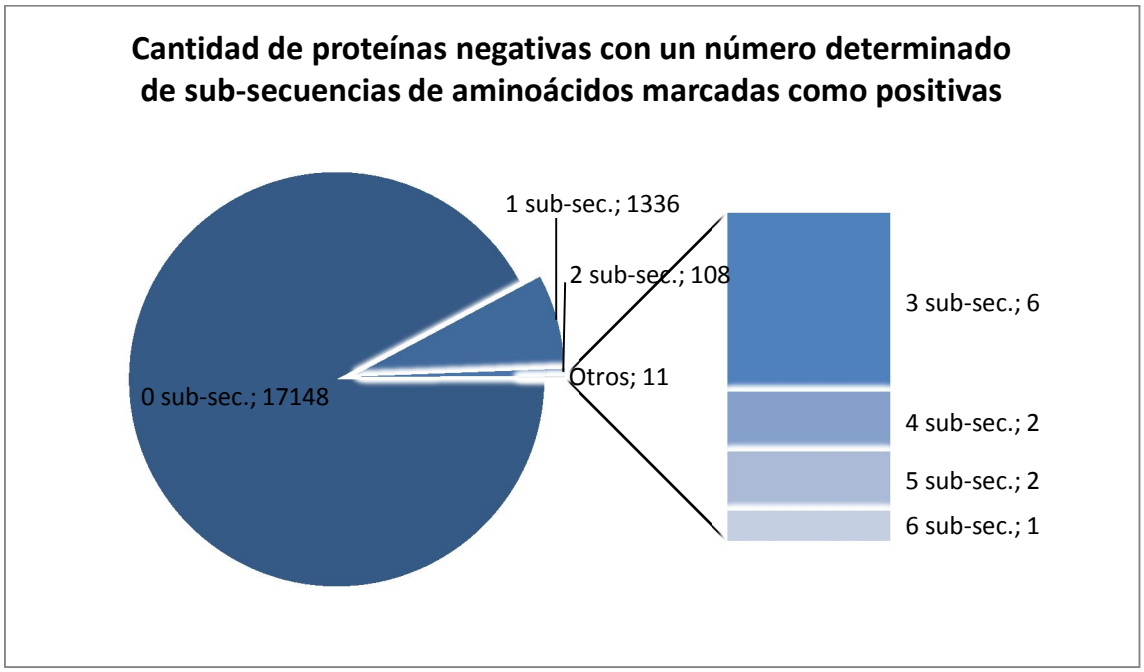
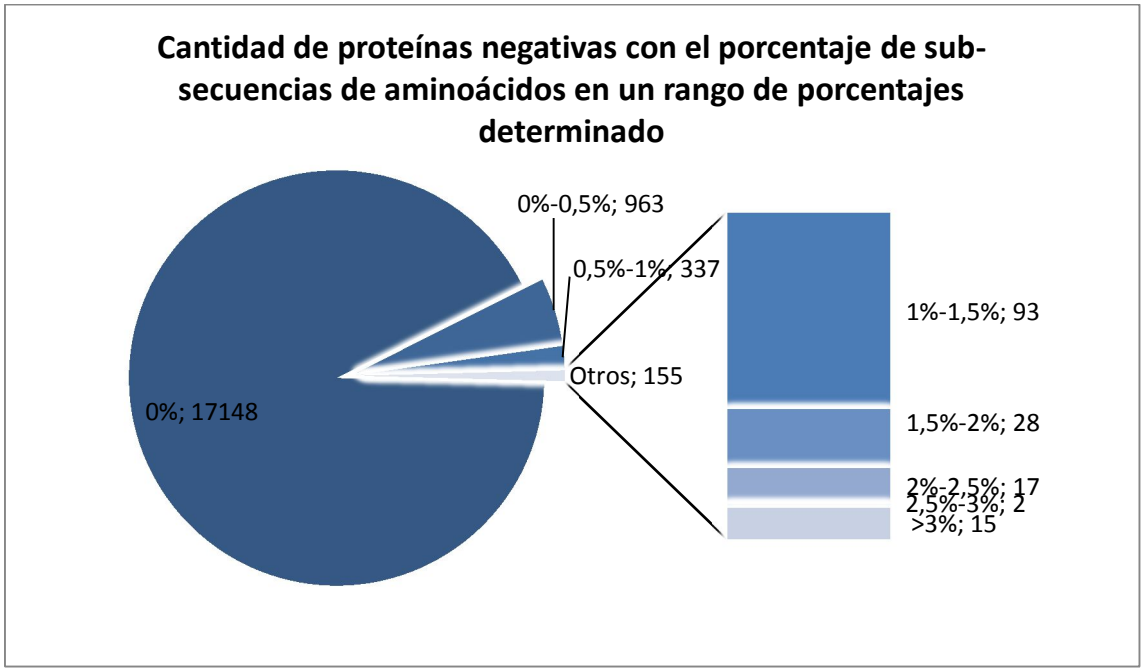


Figura 71: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo Bagging-Part



Este modelo es restrictivo a la hora de clasificar como positivas las sub-secuencias de las proteínas, y sólo 119 proteínas de las 18.603 tienen al menos 2 sub-secuencias marcada como positivas. Las que más destacan son:

Tabla 34: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo Bagging-Part

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
Número total de sub-secuencias clasificadas positivas[24]			
1tr2:B	4	1028	0,39%
2vz8:B	5	2474	0,20%
3cmu:A	6	2012	0,30%
3cmv:A	4	1319	0,30%
3cmw:A	5	1668	0,30%
Mayor porcentaje de sub-secuencias clasificadas positivas sobre el total de sub-secuencias[25]			
1j3w:B	2	125	1,60%
2jqz:A	2	93	2,15%
2oug:A	2	124	1,61%
2zca:B	2	131	1,53%

[24] Con un mínimo de 4 sub-secuencias catalogadas como positivas.
[25] Criterio usado: un mínimo de 2 sub-secuencias catalogadas como positivas y un porcentaje sobre el total mayor o igual al 1,5%.

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 35: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	0	108	0,00%
1ozn:A	1	247	0,40%
1r4x:A	0	237	0,00%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	738	0,00%
2dba:A	0	110	0,00%
2e9g:A	0	93	0,00%
2iv9:A	0	200	0,00%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	397	0,00%
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	104	0,00%
2z5k:A	1	852	0,12%
2zej:B	0	146	0,00%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

4.4.3.10. Modelo Bagging-J48

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 72: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo Bagging-J48

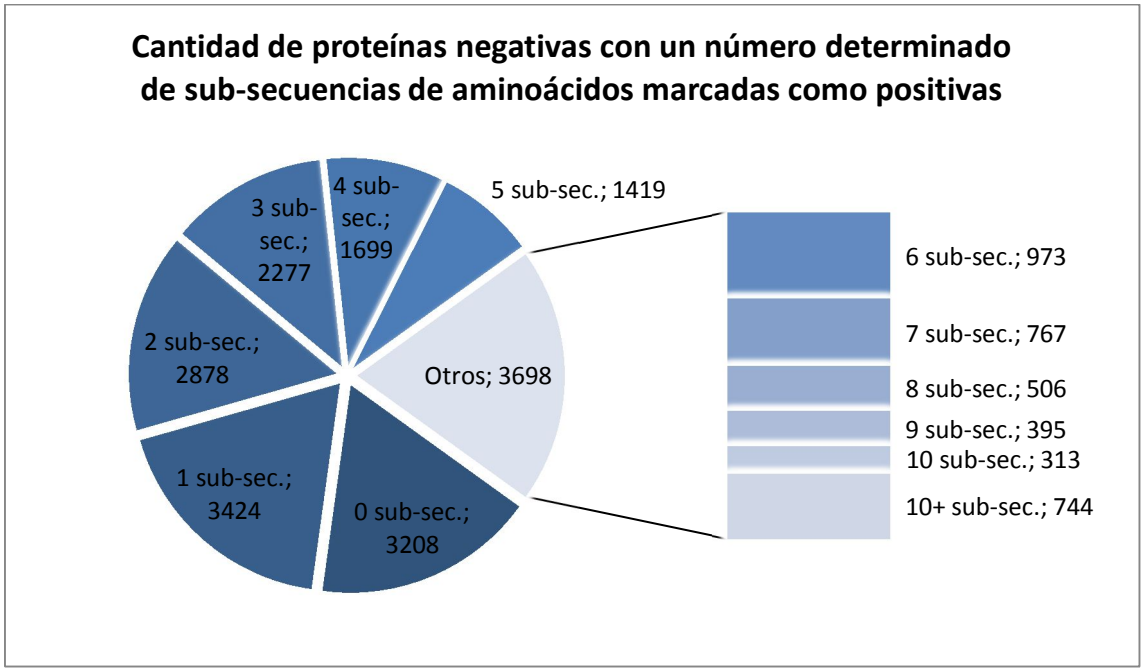
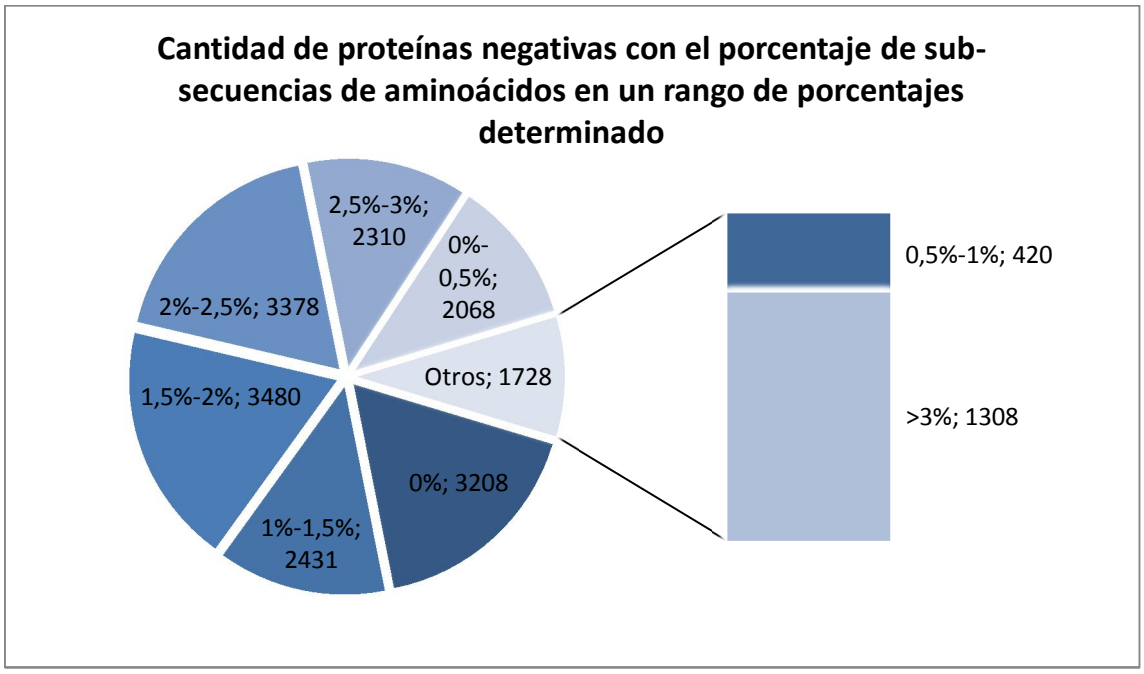


Figura 73: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo Bagging-J48



Debido al reparto tan homogéneo que produce este modelo no destacan algunas proteínas sobre el resto. No obstante, las proteínas que más podrían destacar:

Tabla 36: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo Bagging-J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
Número total de sub-secuencias clasificadas positivas[26]			
1hqm:D	31	1191	2,60%
2be5:N	32	1486	2,15%
2gho:D	31	1195	2,59%
2gsx:A	33	913	3,61%
2q7z:A	66	1893	3,49%
2vz8:B	46	2474	1,86%
Mayor porcentaje de sub-secuencias clasificadas positivas sobre el total de sub-secuencias[27]			
1bcp:F	7	61	11,48%
1ldd:B	5	36	13,89%
1wq6:B	4	34	11,76%
1zza:A	6	52	11,54%
2jt1:A	4	39	10,26%
2pkg:C	5	50	10,00%

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 37: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	0	108	0,00%
1ozn:A	9	247	3,64%
1r4x:A	9	237	3,80%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	10	738	1,36%
2dba:A	2	110	1,82%
2e9g:A	3	93	3,23%
2iv9:A	4	200	2,00%
2vgl:M (la proteína que destacaba en el	6	397	1,51%

[26] Con un mínimo de 31 sub-secuencias catalogadas como positivas.
[27] Criterio usado: un mínimo de 4 sub-secuencias catalogadas como positivas y un porcentaje sobre el total mayor al 10%.

artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)			
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	5	104	4,81%
2z5k:A	18	852	2,11%
2zej:B	3	146	2,05%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

4.4.3.11. Modelo Bagging-NBTree

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 74: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo Bagging-NBTree

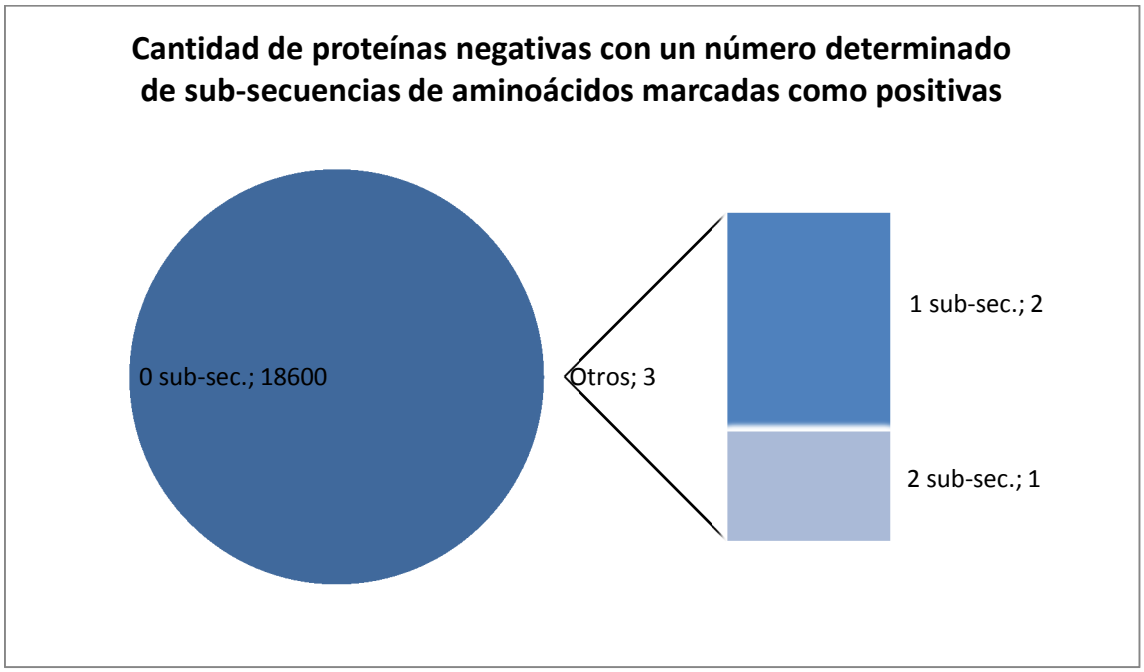
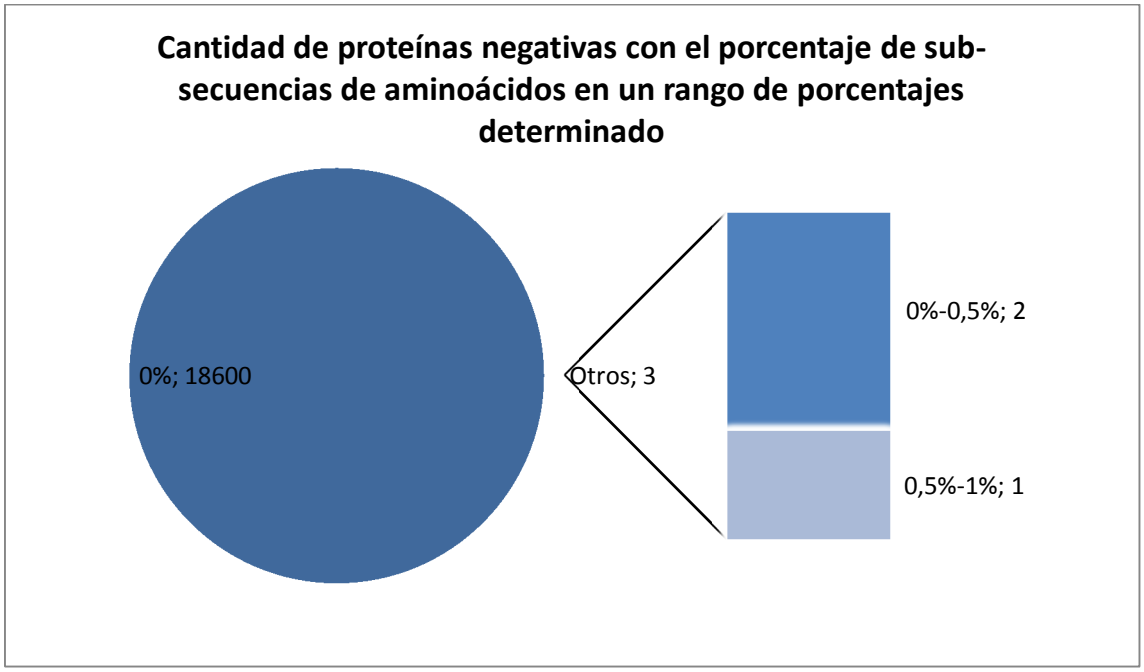


Figura 75: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo Bagging-NBTree



Este modelo es realmente muy restrictivo a la hora de clasificar como positivas las sub-secuencias de las proteínas, y sólo 3 proteínas de las 18.603 tienen alguna sub-secuencia marcada como positiva. Estas 3 proteínas son:

Tabla 38: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo Bagging-NBTree

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
2wtm:A	1	213	0,47%
3l3b:B	1	204	0,49%
2aja:B	2	338	2aja:B

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 39: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	0	108	0,00%
1ozn:A	0	247	0,00%
1r4x:A	0	237	0,00%
1u6g:A (la proteína que destacaba en el	0	738	0,00%

artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)			
2dba:A	0	110	0,00%
2e9g:A	0	93	0,00%
2iv9:A	0	200	0,00%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	397	0,00%
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	104	0,00%
2z5k:A	0	852	0,00%
2zej:B	0	146	0,00%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

4.4.3.12. Modelo RacedIncrementalLogitBoost

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 76: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo RacedIncrementalLogitBoost

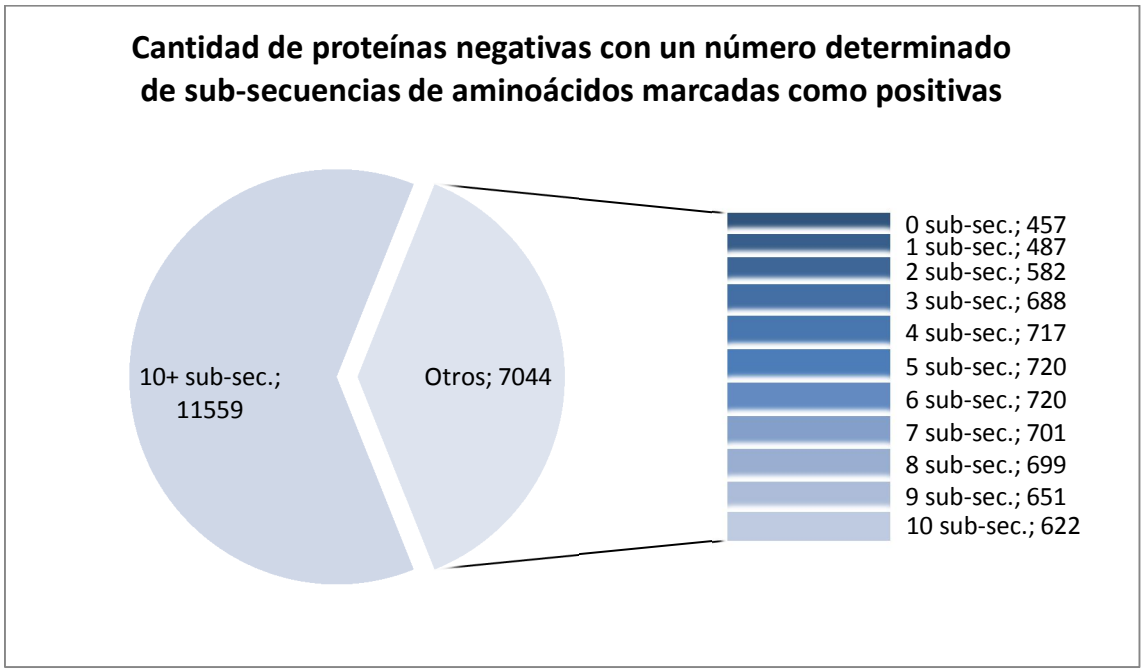
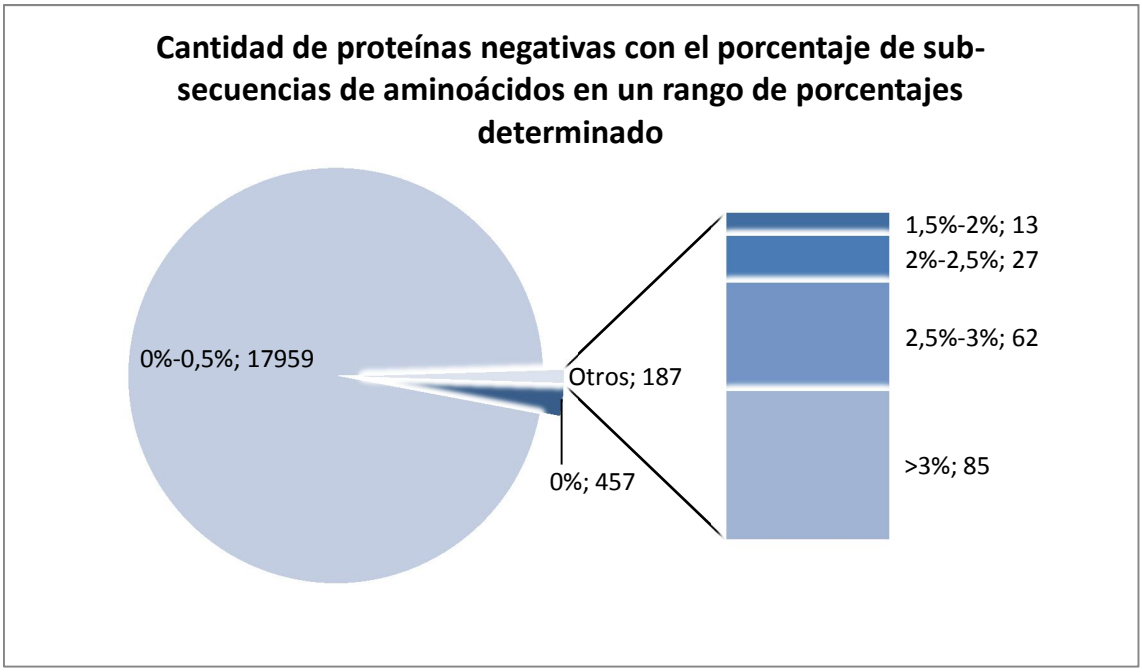


Figura 77: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo



Este modelo cataloga muchas sub-secuencias como positivas. De hecho, más de la mitad de las proteínas tienen más de 10 sub-secuencias catalogadas como negativas. De cara a poder destacar algunas proteínas es necesario aumentar hacer un filtro más estricto:

Tabla 40: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo RacedIncrementalLogitBoost

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
Número total de sub-secuencias clasificadas positivas[28]			
2uv8:B	169	1848	9,15%
2uv8:G	179	2013	8,89%
2vz8:B	244	2474	9,86%
3cmu:A	203	2012	10,09%
3cmw:A	165	1668	9,89%
Mayor porcentaje de sub-secuencias clasificadas positivas sobre el total de sub-secuencias[29]			
1z98:A	38	243	15,64%
2pz9:A	32	188	17,02%
2wyu:C	34	223	15,25%
2ziy:A	52	334	15,57%
3nk6:A	36	239	15,06%

[28] Con un mínimo de 160 sub-secuencias catalogadas como positivas.
[29] Criterio usado: un mínimo de 32 sub-secuencias catalogadas como positivas y un porcentaje sobre el total mayor al 15%.

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 41: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	8	108	7,41%
1ozn:A	24	247	9,72%
1r4x:A	19	237	8,02%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	66	738	8,94%
2dba:A	10	110	9,09%
2e9g:A	7	93	7,53%
2iv9:A	7	200	3,50%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	38	397	9,57%
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	15	104	14,42%
2z5k:A	84	852	9,86%
2zej:B	14	146	9,59%
3ifq:C	3	69	4,35%
3lqv:P	0	1	0,00%

4.4.3.13. Modelo RandomCommittee

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 78: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo RandomCommittee

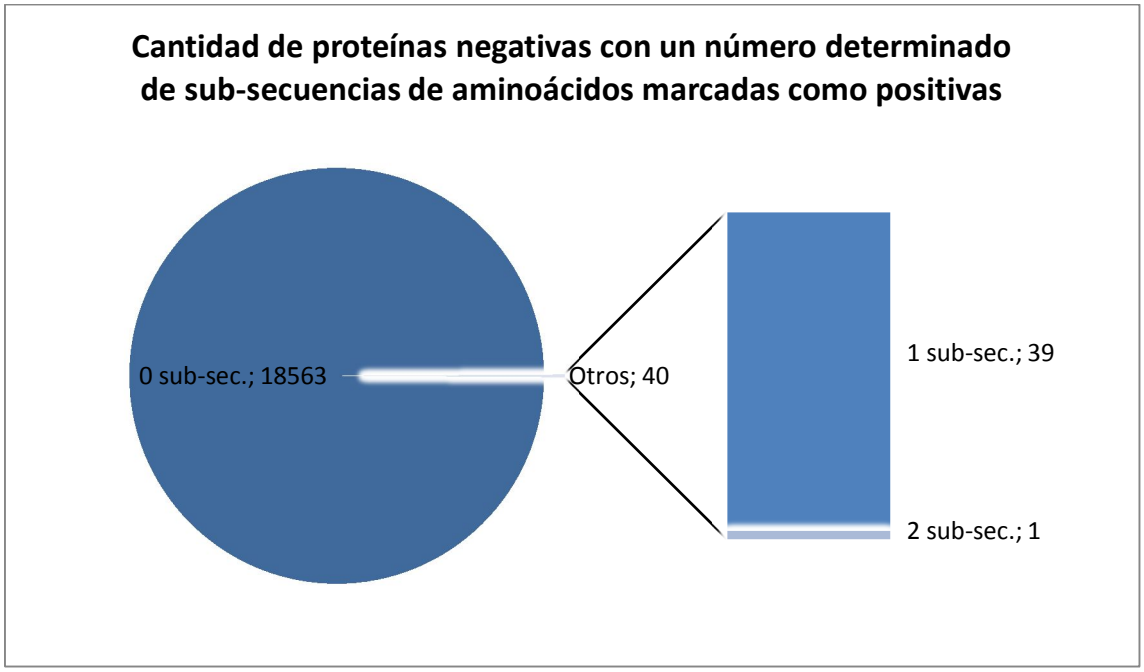
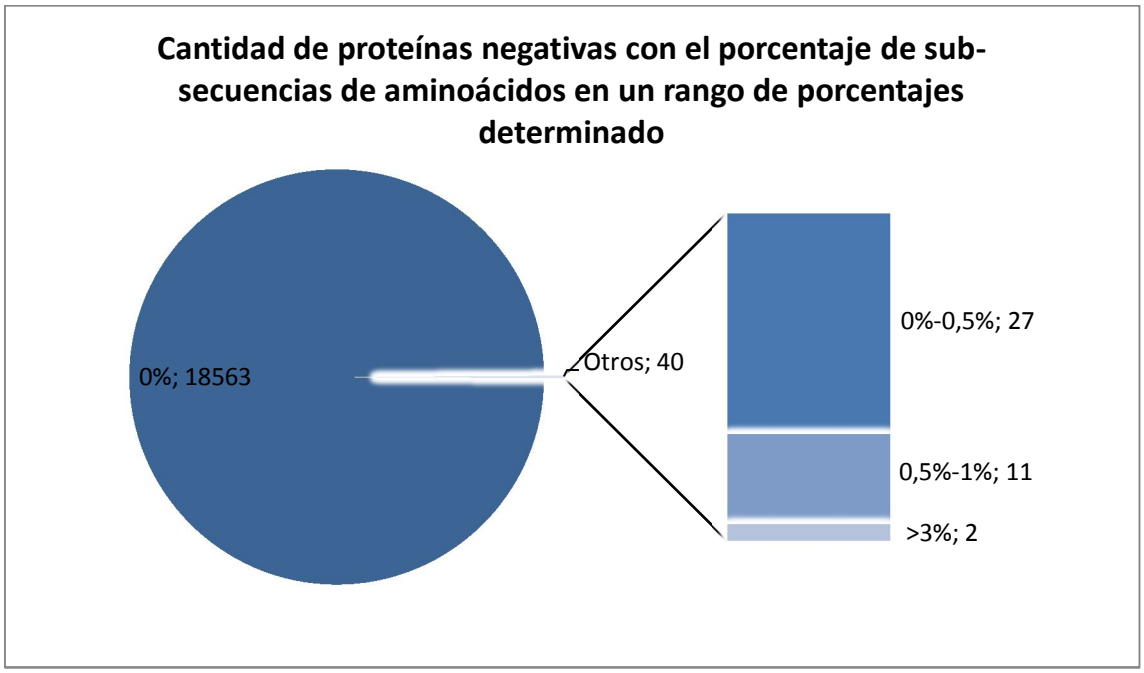


Figura 79: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo RandomCommittee



Este modelo es muy restrictivo a la hora de clasificar como positivas las sub-secuencias de las proteínas, y sólo 40 proteínas de las 18.603 tiene alguna sub-secuencia marcada como positiva, de las cuales sólo 1 tiene 2 sub-secuencias. Destacamos únicamente la proteína con 2 sub-secuencias positivas:

Tabla 42: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo RandomCommittee

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
2aja:B	2	338	0,59%

A modo de pequeña comparación respecto al artículo functional and Genomic Analyses of Alpha-Solenoid Proteins, recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 43: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	0	108	0,00%
1ozn:A	0	247	0,00%
1r4x:A	0	237	0,00%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	738	0,00%
2dba:A	0	110	0,00%
2e9g:A	0	93	0,00%
2iv9:A	0	200	0,00%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	397	0,00%
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	104	0,00%
2z5k:A	0	852	0,00%
2zej:B	0	146	0,00%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

4.4.3.14. Modelo RotationForest-Part

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 80: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo RotationForest-Part

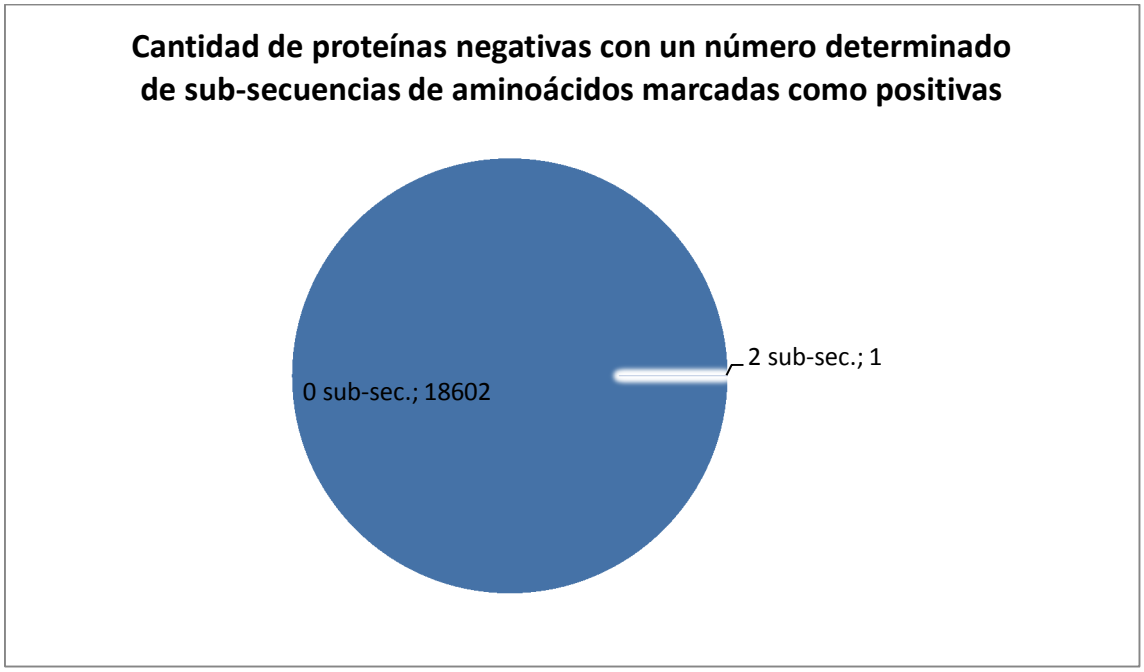
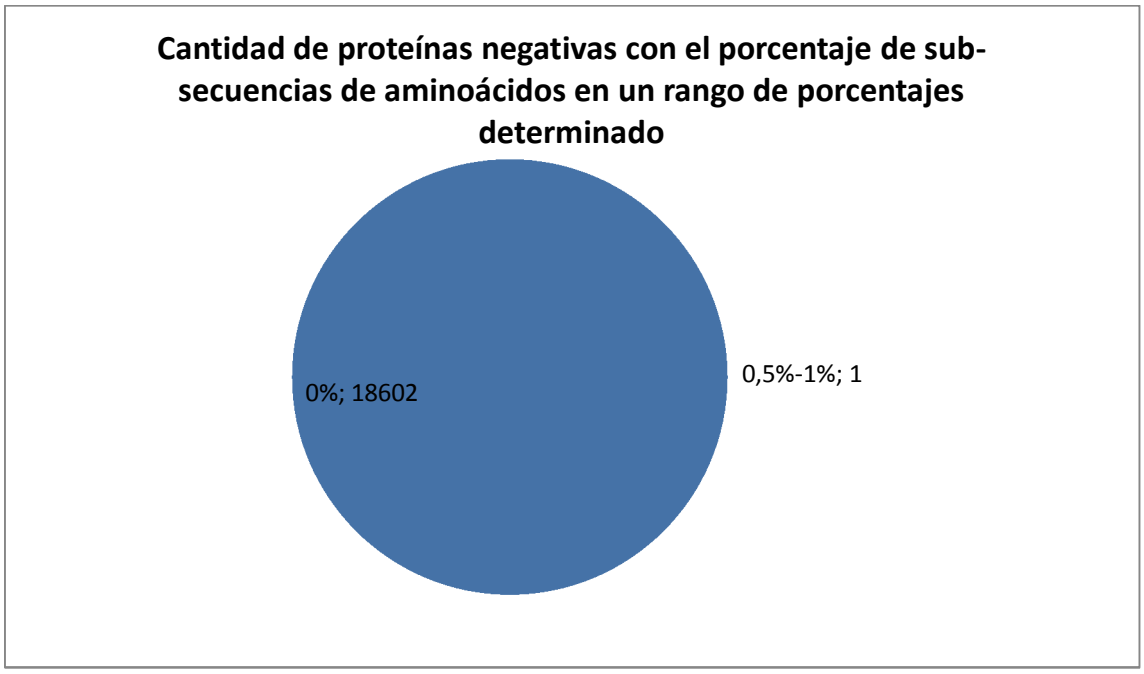


Figura 81: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo RotationForest-Part



Este modelo es tan restrictivo a la hora de clasificar como positivas las sub-secuencias de las proteínas que sólo ha clasificado 2 sub-secuencias como positivas, ambas pertenecientes a la misma proteína:

Tabla 44: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo RotationForest-Part

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
2aja:B	2	338	0,59%

A modo de pequeña comparación respecto al artículo [functional and Genomic Analyses of Alpha-Solenoid Proteins], recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 45: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	0	108	0,00%
1ozn:A	0	247	0,00%
1r4x:A	0	237	0,00%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	738	0,00%
2dba:A	0	110	0,00%
2e9g:A	0	93	0,00%
2iv9:A	0	200	0,00%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	397	0,00%
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	104	0,00%
2z5k:A	0	852	0,00%
2zej:B	0	146	0,00%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

4.4.3.15. Modelo RotationForest-J48

Con los siguientes gráficos se puede ver cómo ha clasificado las proteínas negativas este modelo:

Figura 82: Gráfica con el número de proteínas a priori negativas con un número determinado de sub-secuencias de aminoácidos marcados como positivas, con el modelo RotationForest-J48

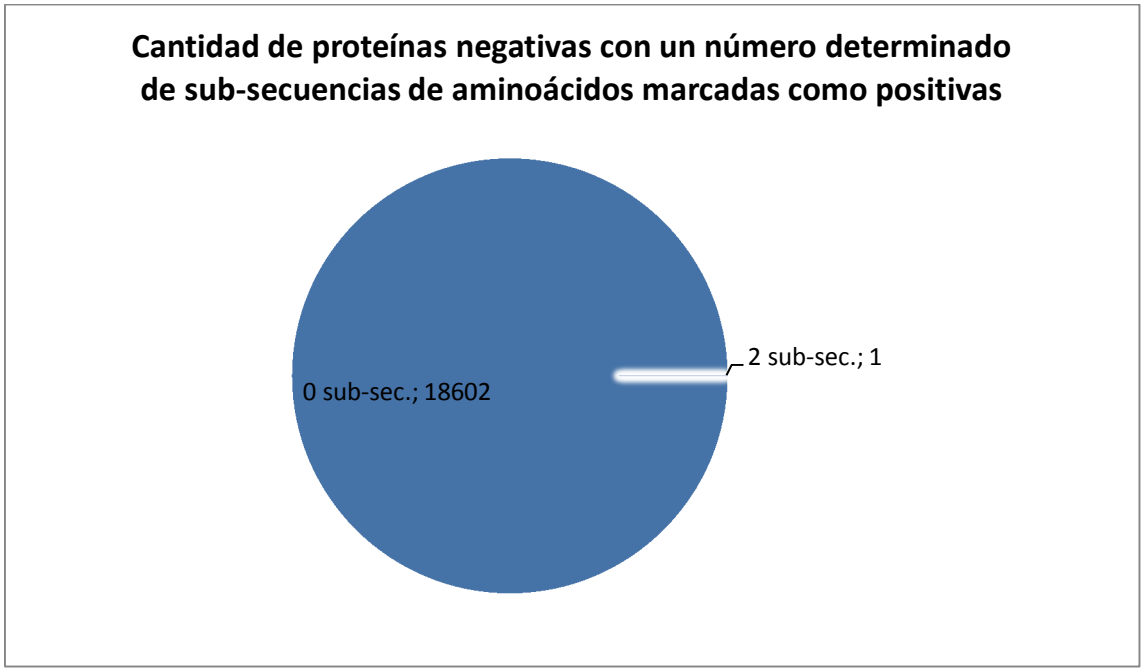
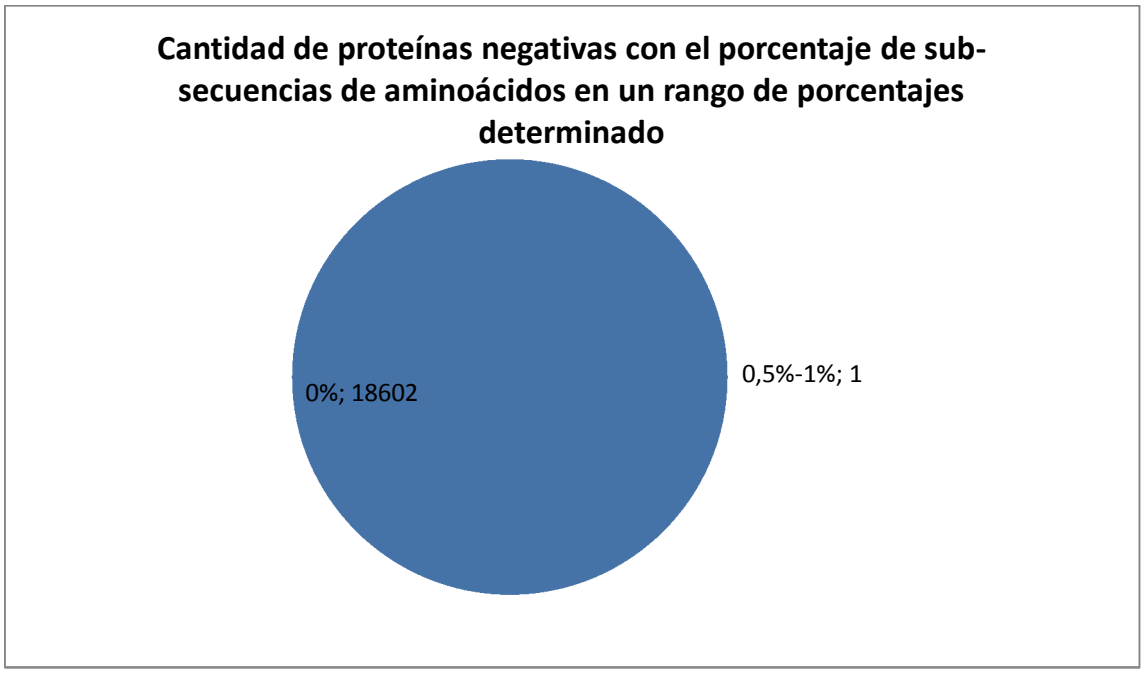


Figura 83: Gráfica que muestra la cantidad de proteínas negativas que tienen un porcentaje de sub-secuencias de aminoácidos marcadas como positivas en un rango determinado, con la catalogación que ha hecho el modelo RotationForest-J48



Este modelo es tan restrictivo a la hora de clasificar como positivas las sub-secuencias de las proteínas que sólo ha clasificado 2 sub-secuencias como positivas, ambas pertenecientes a la misma proteína:

Tabla 46: Proteínas inicialmente negativas que más destacan para poder ser en realidad positivas con el modelo RotationForest-J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
2aja:B	2	338	0,59%

A modo de pequeña comparación respecto al artículo [functional and Genomic Analyses of Alpha-Solenoid Proteins], recopilamos las proteínas que ahí destacaban por algún motivo (tabla S3 del mismo) para ver cómo se comportan bajo este modelo:

Tabla 47: Proteínas inicialmente negativas que destacaban en el artículo predecesor a este trabajo, con la cantidad de sub-secuencias que salen como positivas en este trabajo, usando el modelo J48

Proteína	Número sub-secuencias positivas	Número sub-secuencias en total	Porcentaje sub-secuencias positivas
1iu1:B	0	108	0,00%
1ozn:A	0	247	0,00%
1r4x:A	0	237	0,00%
1u6g:A (la proteína que destacaba en el artículo era la 1U6G:C, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	738	0,00%
2dba:A	0	110	0,00%
2e9g:A	0	93	0,00%
2iv9:A	0	200	0,00%
2vgl:M (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	397	0,00%
2vgl:S (la proteína que destacaba en el artículo era la 2vgl:B, pero dicha proteína no está en el juego de datos que manejamos y, por similitud, hemos optado por incluir esta)	0	104	0,00%
2z5k:A	0	852	0,00%
2zej:B	0	146	0,00%
3ifq:C	0	69	0,00%
3lqv:P	0	1	0,00%

5. Resultados obtenidos

A continuación vamos a recopilar, agrupar e interpretar los datos obtenidos durante la experimentación. Los puntos en los que se divide esta tarea:

1. Por un lado vamos a determinar, con diferentes algoritmos, qué atributos tienen más importancia, dentro de las sub-secuencias de 39 aminoácidos, para determinar la clase (positiva/negativa) de la sub-secuencia.
2. Con los resultados de la generación de cada modelo con el fichero de entrenamiento vamos a comparar cada uno de los modelos por su porcentaje de clasificación de sub-secuencias negativas y así poder ver qué modelos clasifican mejor las sub-secuencias del fichero de entrenamiento ya que, a priori, deberían ser también los que mejor clasifiquen las sub-secuencias de los ficheros de proteínas positivas y negativas.
3. Agrupación y comparación de resultados al aplicar los modelos sobre el fichero de proteínas positivas.
4. Agrupación y comparación de resultados al aplicar los modelos sobre el fichero de proteínas negativas.
5. Resumen de las proteínas inicialmente negativas que más probabilidad tienen de ser falsos positivos a tenor de los resultados obtenidos sobre las sub-secuencias de todas esas proteínas y con la información sobre la fiabilidad de cada uno de los modelos.
6. También se provee una tabla resumen con las proteínas que destacaban en el artículo previo a este trabajo, con los resultados que dichas proteínas obtienen en este trabajo, de forma que se pueda hacer una pequeña comparación de los resultados de dicho artículo con este trabajo.

5.1. Selección de atributos por su importancia

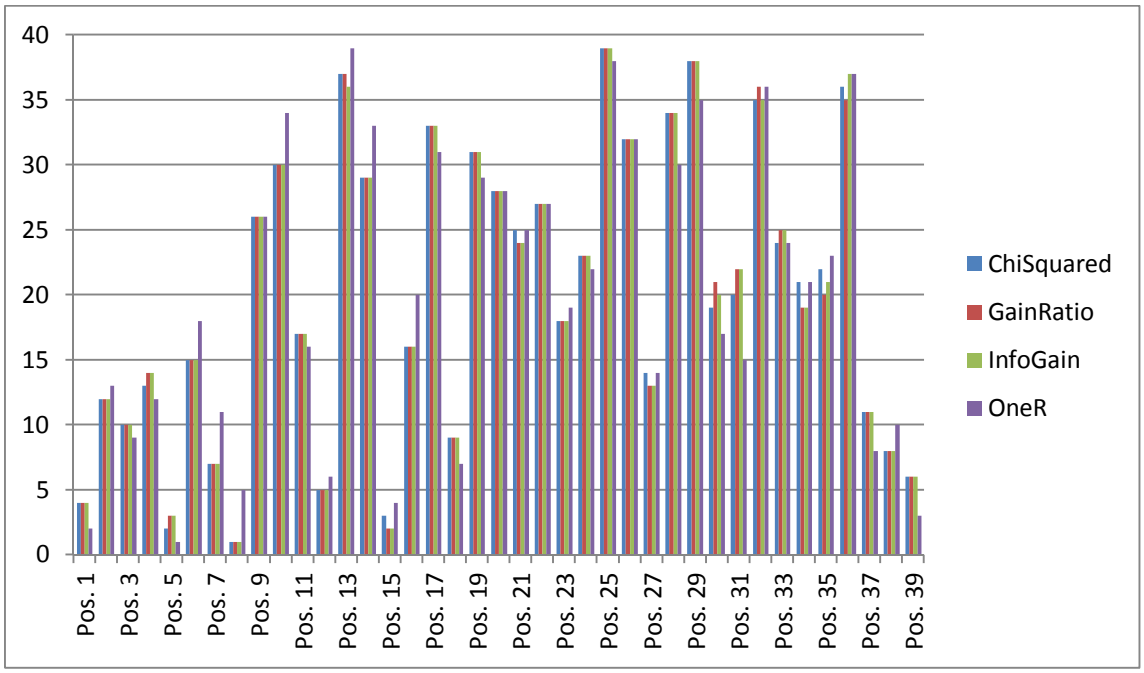
A partir del Fichero de entrenamiento para Weka, base para la generación de modelos vamos a obtener un ranking de las posiciones de las sub-secuencias de aminoácidos según su importancia para determinar que la sub-secuencia es de clase positiva, según el evaluador de atributos. Es decir, vamos a usar diferentes algoritmos para determinar el orden de importancia de cada una de posiciones de las sub-secuencias de cara a determinar la clase de la sub-secuencia:

Tabla 48: Ranking de atributos (ordenados de mayor a menor peso) que determinan si una sub-secuencia es positiva en el modelo PART

	ChiSquared	GainRatio	InfoGain	OneR
1	Pos. 25	Pos. 25	Pos. 25	Pos. 13
2	Pos. 29	Pos. 29	Pos. 29	Pos. 25
3	Pos. 13	Pos. 13	Pos. 36	Pos. 36
4	Pos. 36	Pos. 32	Pos. 13	Pos. 32
5	Pos. 32	Pos. 36	Pos. 32	Pos. 29
6	Pos. 28	Pos. 28	Pos. 28	Pos. 10
7	Pos. 17	Pos. 17	Pos. 17	Pos. 14
8	Pos. 26	Pos. 26	Pos. 26	Pos. 26
9	Pos. 19	Pos. 19	Pos. 19	Pos. 17
10	Pos. 10	Pos. 10	Pos. 10	Pos. 28
11	Pos. 14	Pos. 14	Pos. 14	Pos. 19
12	Pos. 20	Pos. 20	Pos. 20	Pos. 20
13	Pos. 22	Pos. 22	Pos. 22	Pos. 22
14	Pos. 9	Pos. 9	Pos. 9	Pos. 9
15	Pos. 21	Pos. 33	Pos. 33	Pos. 21
16	Pos. 33	Pos. 21	Pos. 21	Pos. 33
17	Pos. 24	Pos. 24	Pos. 24	Pos. 35
18	Pos. 35	Pos. 31	Pos. 31	Pos. 24
19	Pos. 34	Pos. 30	Pos. 35	Pos. 34
20	Pos. 31	Pos. 35	Pos. 30	Pos. 16
21	Pos. 30	Pos. 34	Pos. 34	Pos. 23
22	Pos. 23	Pos. 23	Pos. 23	Pos. 6
23	Pos. 11	Pos. 11	Pos. 11	Pos. 30
24	Pos. 16	Pos. 16	Pos. 16	Pos. 11
25	Pos. 6	Pos. 6	Pos. 6	Pos. 31
26	Pos. 27	Pos. 4	Pos. 4	Pos. 27
27	Pos. 4	Pos. 27	Pos. 27	Pos. 2
28	Pos. 2	Pos. 2	Pos. 2	Pos. 4
29	Pos. 37	Pos. 37	Pos. 37	Pos. 7
30	Pos. 3	Pos. 3	Pos. 3	Pos. 38
31	Pos. 18	Pos. 18	Pos. 18	Pos. 3
32	Pos. 38	Pos. 38	Pos. 38	Pos. 37
33	Pos. 7	Pos. 7	Pos. 7	Pos. 18
34	Pos. 39	Pos. 39	Pos. 39	Pos. 12
35	Pos. 12	Pos. 12	Pos. 12	Pos. 8
36	Pos. 1	Pos. 1	Pos. 1	Pos. 15
37	Pos. 15	Pos. 5	Pos. 5	Pos. 39
38	Pos. 5	Pos. 15	Pos. 15	Pos. 1
39	Pos. 8	Pos. 8	Pos. 8	Pos. 5

Estos datos los vamos a llevar a un gráfico para verlos mejor. En dicho gráfico el eje vertical indica la importancia de la posición según el algoritmo de selección, siendo el máximo valor 39 (posición más importante para ese algoritmo) y el 1 el valor mínimo (posición menos importante):

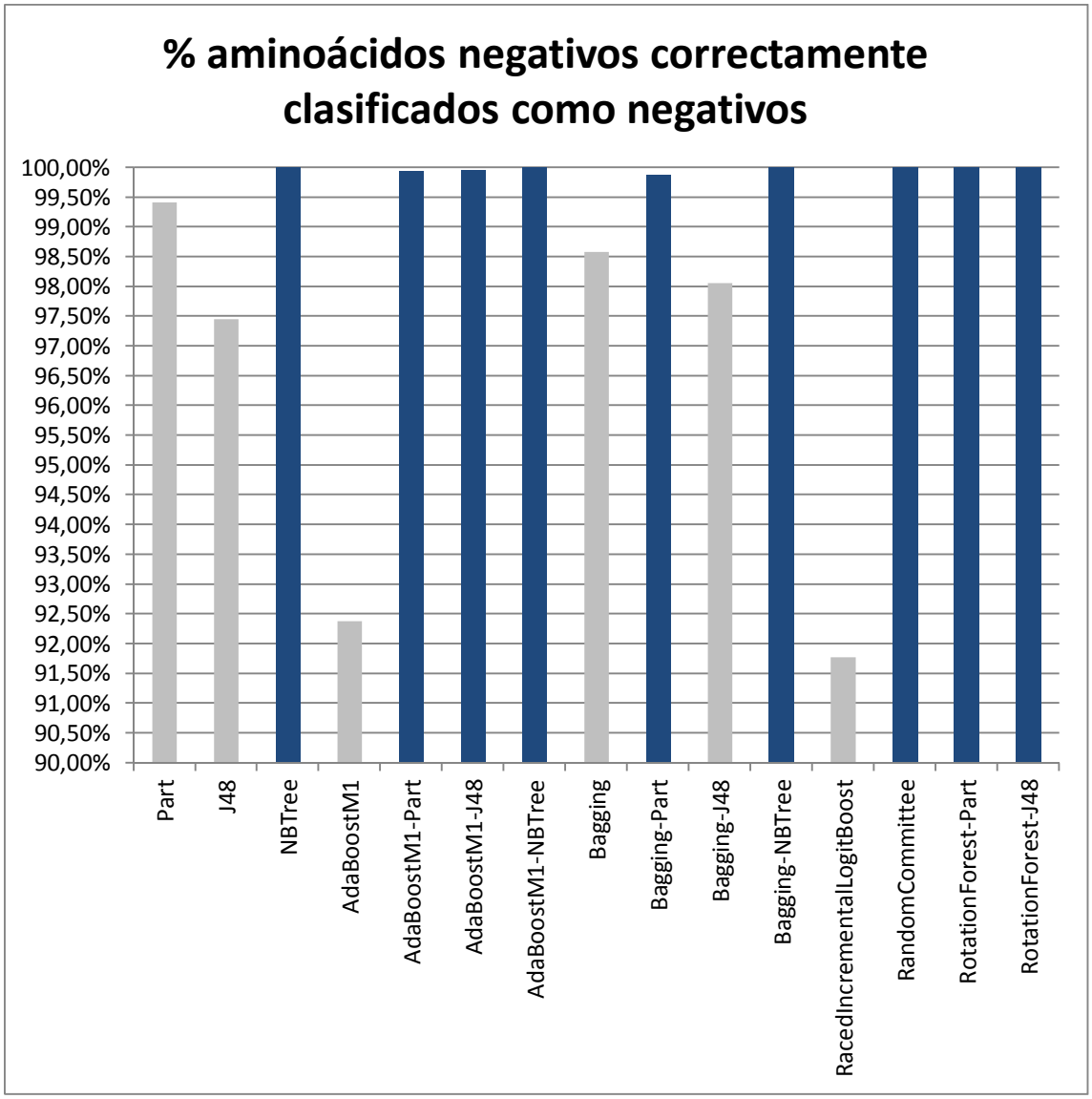
Figura 84: Gráfica de importancia de las posiciones de las cadenas de aminoácidos, según algoritmo de selección



5.2.Comparativa de modelos según la clasificación de sub-secuencias de aminoácidos negativos con el fichero de entrenamiento

Cogiendo el porcentaje de sub-secuencias de aminoácidos negativas clasificadas correctamente como negativos en la generación de cada uno de los modelos, a partir del fichero de entrenamiento, del punto modelos y reglas se ha hecho el siguiente gráfico, cuyo propósito es ver de forma rápida, una comparativa de dicho porcentaje por modelo, y así ver cuáles son, en principio, mejores evaluadores:

Figura 85: Tabla de comparativa de porcentaje de sub-secuencias de aminoácidos negativos clasificados correctamente por cada uno de los modelos



Nota: en azul los modelos que superan el 99,5% de acierto de clasificación de sub-secuencias negativas del fichero de entrenamiento

Los modelos que más destacan por su mayor porcentaje de clasificación correcta de sub-secuencias de aminoácidos negativos (más del 99,5% de acierto) son:

- NBTree
- AdaBoostM1-Part
- AdaBoostM1-J48
- AdaBoostM1-NBTree
- Bagging-Part
- Bagging-NBTree
- RandomCommittee
- RotationForest-Part
- RotationForest-J48

Nota: estos modelos que destacan habían catalogado el 100% de las sub-secuencias positivas correctamente en el fichero de entrenamiento. Aunque la clasificación de las sub-secuencias positivas no es determinante puesto que en el fichero de entrenamiento se tuvo que repetir varias veces las sub-secuencias positivas originales, por lo que al haber el mismo conjunto de datos repetidas veces es más sencillo de catalogar.

5.3.Comparativa de los modelos en función del número de proteínas positivas con la cantidad de sub-secuencias marcadas como positivas

Cogiendo los datos del apartado aplicación de los modelos generados sobre el fichero de proteínas positivas y agrupándolos en una tabla resumen:

Tabla 49: número de proteínas, del fichero de proteínas positivas, con un determinado número de sub-secuencias marcadas como positivas según cada uno de los modelos usados

	Número de proteínas con el total de sub-secuencias catalogadas como positivas											
	0	1	2	3	4	5	6	7	8	9	10	10+
Part	17	21	18	20	20	6	7	3	4	5	5	3
J48	3	2	3	5	10	15	7	7	6	2	8	61
NBTree	113	10	0	1	1	0	0	2	0	0	0	2
AdaBoostM1	0	0	2	1	0	2	1	0	1	0	2	120
AdaBoostM1-Part	96	13	10	2	2	0	0	2	2	0	0	2
AdaBoostM1-J48	96	20	4	2	1	2	0	2	0	0	0	2
AdaBoostM1-NBTree	113	10	0	1	1	0	0	2	0	0	0	2
Bagging	3	6	10	12	14	9	12	10	10	6	6	31
Bagging-Part	67	39	9	3	3	1	0	4	0	0	1	2
Bagging-J48	3	3	4	13	11	15	7	7	8	9	4	45
Bagging-NBTree	122	1	0	1	1	0	0	2	0	0	0	2
RacedIncrementalLogitBoost	0	0	0	0	0	1	2	0	2	0	2	122
RandomCommittee	123	0	0	1	1	0	0	2	0	0	0	2
RotationForest-Part	123	0	0	1	1	0	0	2	0	0	0	2
RotationForest-J48	121	2	0	1	1	0	0	2	0	0	0	2

Según lo que se había visto en el punto comparativa de clasificaciones de sub-secuencias de aminoácidos negativos según modelo, los modelos que mejor porcentaje de marcaje de sub-secuencias de aminoácidos negativos eran:

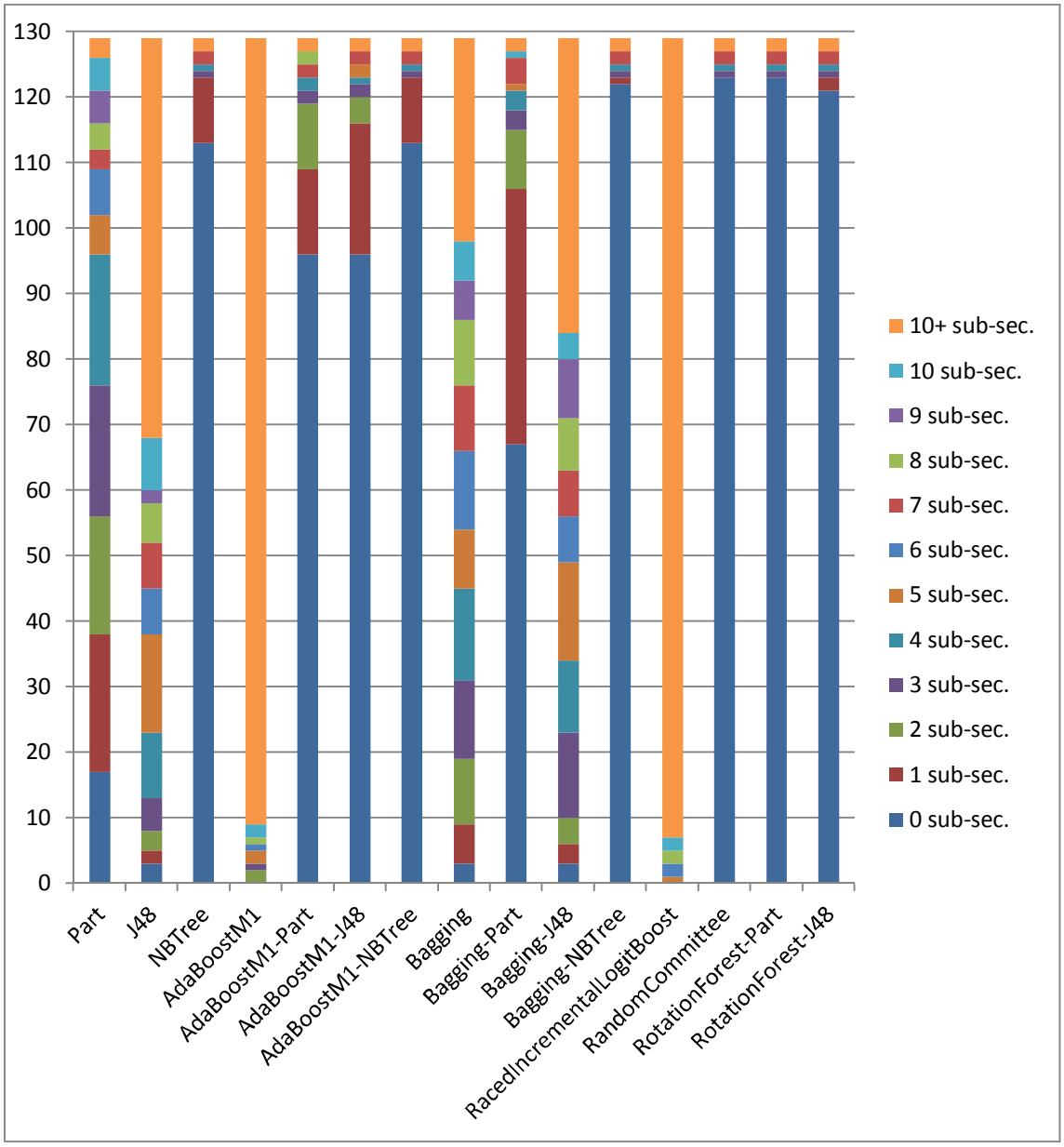
- NBTree
- AdaBoostM1-Part
- AdaBoostM1-J48
- AdaBoostM1-NBTree
- Bagging-Part
- Bagging-NBTree
- RandomCommittee
- RotationForest-Part
- RotationForest-J48

Según podemos ver en la tabla anterior, los resultados sobre las proteínas positivas con estos modelos se parecen mucho entre sí, predominando claramente la detección de 0 sub-secuencias de aminoácidos positivas en las proteínas, por lo que parece que son modelos muy restrictivos a la hora de detectar sub-secuencias positivas. Por ello, en principio lo que

cabe esperar es que la aplicación de estos modelos sobre el fichero de proteínas inicialmente negativas nos den pocas proteínas candidatas a ser positivas.

Para hacer estos resultados más visibles los convertimos en un gráfico, donde se puede ver de un vistazo una comparativa de los resultados de todos los modelos, mostrando el número de proteínas que tienen un determinado número de sub-secuencias:

Figura 86: Gráfica que muestra el número de proteínas, del fichero de proteínas positivas, con un determinado número de sub-secuencias marcadas como positivas según cada uno de los modelos usados



5.4. Comparativa de los modelos en función del número de proteínas negativas con la cantidad de sub-secuencias marcadas como positivas

Cogiendo los datos del apartado aplicación de los modelos generados sobre el fichero de proteínas negativas y agrupándolos en una tabla resumen:

Tabla 50: número de proteínas, del fichero de proteínas negativas, con un determinado número de sub-secuencias marcadas como positivas según cada uno de los modelos usados

	Número de proteínas con el total de sub-secuencias catalogadas como positivas											
	0	1	2	3	4	5	6	7	8	9	10	10+
Part	9732	5014	2239	949	406	152	48	35	13	10	2	3
J48	2172	2549	2429	2124	1751	1481	1200	1027	807	631	542	1890
NBTree	17221	1202	147	21	8	3	0	0	1	0	0	0
AdaBoostM1	957	915	927	935	973	880	891	850	732	715	657	9171
AdaBoostM1-Part	18442	156	5	0	0	0	0	0	0	0	0	0
AdaBoostM1-J48	18209	383	8	0	1	1	1	0	0	0	0	0
AdaBoostM1-NBTree	17221	1202	147	21	8	3	0	0	1	0	0	0
Bagging	4028	3753	2994	2218	1634	1141	850	564	439	280	199	503
Bagging-Part	17148	1336	108	6	2	2	1	0	0	0	0	0
Bagging-J48	3208	3424	2878	2277	1699	1419	973	767	506	395	313	744
Bagging-NBTree	18600	2	1	0	0	0	0	0	0	0	0	0
RacedIncrementalLogitBoost	457	487	582	688	717	720	720	701	699	651	622	11559
RandomCommittee	18563	39	1	0	0	0	0	0	0	0	0	0
RotationForest-Part	18602	0	1	0	0	0	0	0	0	0	0	0
RotationForest-J48	18602	0	1	0	0	0	0	0	0	0	0	0

Según lo que se había visto en el punto comparativa de clasificaciones de sub-secuencias de aminoácidos negativos según modelo, los modelos que mejor porcentaje de marcaje de sub-secuencias de aminoácidos negativos eran:

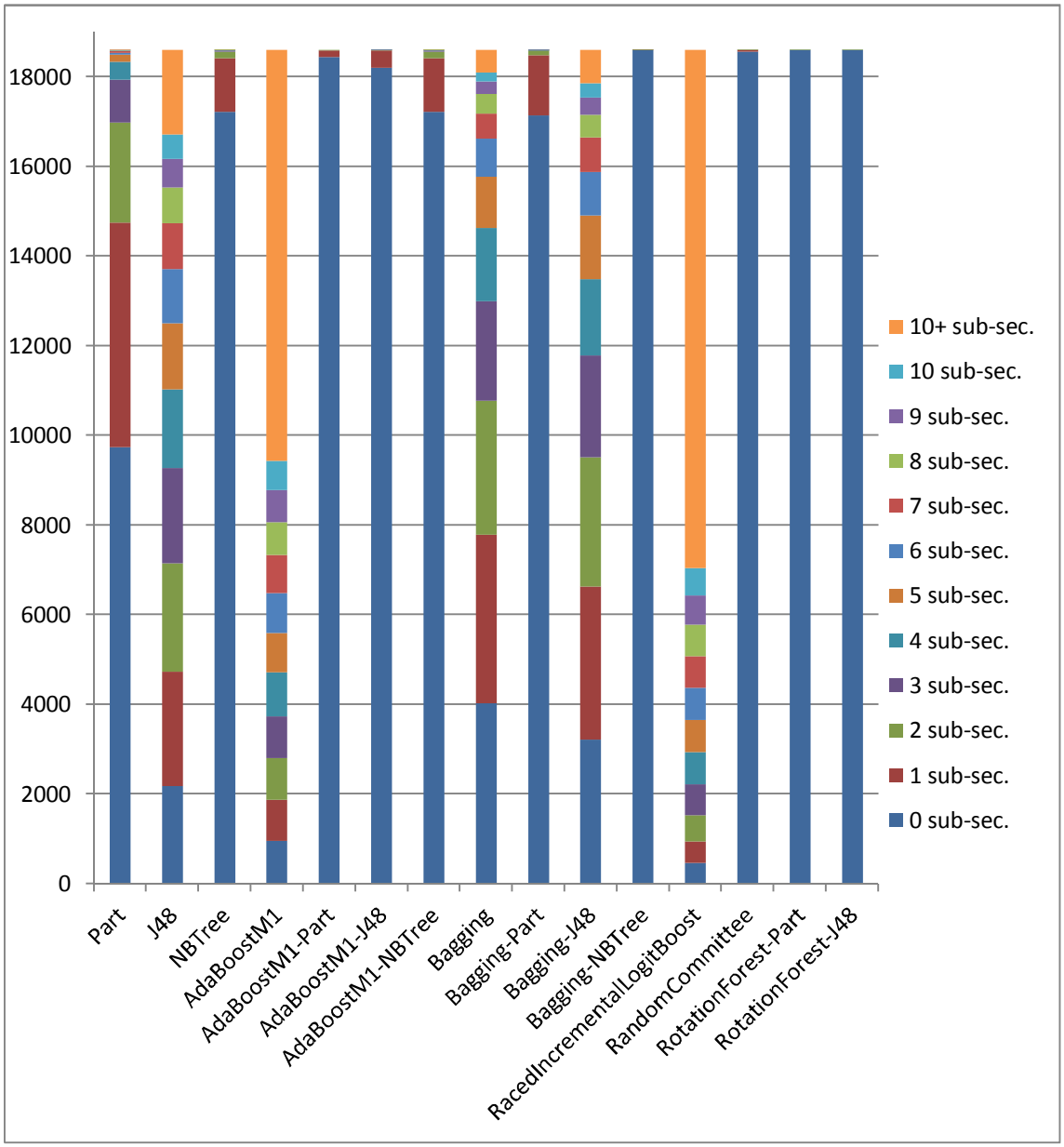
- NBTree
- AdaBoostM1-Part
- AdaBoostM1-J48
- AdaBoostM1-NBTree
- Bagging-Part
- Bagging-NBTree
- RandomCommittee
- RotationForest-Part
- RotationForest-J48

Según podemos ver en la tabla anterior, los resultados sobre las proteínas negativas con estos modelos se parecen entre sí, predominando claramente la detección de 0 sub-secuencias de aminoácidos positivas en las proteínas, tal como ocurría con las proteínas

positivas. Lo que esperamos con dichos resultados tan restrictivos de estos modelos destacados es que detecten pocas proteínas candidatas a ser positivas aplicando un filtro más difícil de pasar.

Para hacer estos resultados más visibles los convertimos en un gráfico, donde se puede ver de un vistazo una comparativa de los resultados de todos los modelos, mostrando el número de proteínas que tienen un determinado número de sub-secuencias:

Figura 87: Gráfica que muestra el número de proteínas, del fichero de proteínas negativas, con un determinado número de sub-secuencias marcadas como positivas según cada uno de los modelos usados



5.5.Proteínas negativas con más probabilidad de ser positivas

Durante la aplicación de los modelos generados sobre el fichero de proteínas negativas habían destacado algunas proteínas en cada uno de los modelos. Las proteínas que se habían destacado eran o por número total de sub-secuencias marcadas como positivas (independientemente del número de sub-secuencias que tuviera cada proteína) como por el porcentaje de sub-secuencias que se habían marcado como positivas sobre el total de sub-secuencias que tuviera la proteína. Por ello, para poder destacar las proteínas vamos a seguir el mismo criterio y se van a destacar en función del criterio seguido:

5.5.1. Por número total de sub-secuencias marcadas como positivas

En la tabla de a continuación aparecen todas las proteínas que han destacado en alguno de los modelos:

Tabla 51: proteínas negativas que destacan por el número total de sub-secuencias de aminoácidos clasificados como positivos, en la aplicación de cada uno de los modelos generados sobre el fichero de proteínas negativas

	Part	J48	NBTree	AdaBoostM1	AdaBoostM1-Part	AdaBoostM1-J48	AdaBoostM1-NBTree	Bagging	Bagging-Part	Bagging-J48	Bagging-NBTree	RacedIncrementalLogitBoost	RandomCommittee	RotationForest-Part	RotationForest-J48
1aud:A						✓									
1cx0:A						✓									
1cza:N	✓														
1dx1:A			✓				✓								
1hqm:D										✓					
1m1c:A						✓									
1t08:A					✓										
1tr2:B									✓						
1w36:C						✓									
1w36:E				✓											
1wa5:B						✓									
2a3l:A					✓										
2aja:B					✓	✓					✓		✓	✓	✓
2be5:N										✓					
2bji:A						✓									
2eyq:A				✓											
2gho:D										✓					
2gsx:A		✓						✓		✓					
2q7z:A		✓						✓		✓					
2qr4:A					✓										
2uv8:B												✓			
2uv8:G												✓			

	Part	J48	NBTree	AdaBoostM1	AdaBoostM1-Part	AdaBoostM1-J48	AdaBoostM1-NBTree	Bagging	Bagging-Part	Bagging-J48	Bagging-NBTree	RacedIncrementalLogitBoost	RandomCommittee	RotationForest-Part	RotationForest-J48
2uvo:F			✓				✓								
2vz8:B	✓	✓		✓				✓	✓	✓		✓			
2waq:B	✓														
2wtm:A											✓				
2x2h:D			✓				✓								
2z5k:A					✓										
3cmu:A	✓			✓		✓			✓			✓			
3cmv:A						✓			✓						
3cmw:A	✓					✓			✓			✓			
3cu7:A								✓							
3ecq:B		✓													
3gau:A			✓				✓	✓							
3haz:A				✓											
3l3b:B											✓				
3pgw:P						✓									

En la tabla se han sombreado de gris las columnas correspondientes a los modelos que mejor porcentaje de clasificación de sub-secuencias de aminoácidos negativos dieron, tal como se puede ver en la comparativa de modelos según la clasificación de sub-secuencias de aminoácidos negativos con el fichero de entrenamiento. Estos se considerarán como modelos *buenos*, mientras que el resto se considerarán como modelos *malos*.

Como se puede ver en la tabla, hay varias proteínas que destacan en varios modelos:

Tabla 52: tabla resumen con las proteínas negativas que destacan por el número total de sub-secuencias de aminoácidos clasificados como positivos, en más de un modelo tras la aplicación de cada uno de los modelos generados sobre el fichero de proteínas negativas

Proteína	Cantidad de modelos *buenos* en los que destaca	Cantidad de modelos *malos* en los que destaca
1dx1:A	2	0
2aja:B	6	0
2gsx:A	0	3
2q7z:A	0	3
2uvo:F	2	0
2vz8:B	1	6
2x2h:D	2	0
3cmu:A	2	3
3cmv:A	2	0
3cmw:A	2	2
3gau:A	2	1

Con este tipo de clasificación hay varias proteínas que destacan en más de 3 modelos aplicados, por lo que parece más fiable de cara a detectar posibles proteínas negativas erróneamente clasificadas.

5.5.2. Por porcentaje de sub-secuencias marcadas como positivas sobre el total de sub-secuencias

	Part	J48	NBTree	AdaBoostM1	AdaBoostM1-Part	AdaBoostM1-J48	AdaBoostM1-NBTree	Bagging	Bagging-Part	Bagging-J48	Bagging-NBTree	RacedIncrementalLogitBoost	RandomCommittee	RotationForest-Part	RotationForest-J48
1aud:A						✓									
1b28:A						✓									
1bcp:F		✓								✓					
1cx0:A						✓									
1dx1:A			✓				✓								
1f2h:A	✓														
1ghh:A					✓										
1h8g:A		✓													
1hcc:A								✓							
1j3w:B									✓						
1kg1:A						✓									
1koy:A								✓							
1kve:C								✓							
1ldd:B										✓					
1m56:J								✓							
1nla:A						✓									
1opi:A		✓													
1syx:B								✓							
1wq6:B										✓					
1y3k:A						✓									
1vaz:A								✓							
1z98:A												✓			
1zza:A		✓								✓					
2bf3:A					✓										
2ct2:A						✓									
2g1u:A	✓														
2ge7:B		✓													
2i8b:B	✓														
2iih:A	✓														

	Part	J48	NBTree	AdaBoostM1	AdaBoostM1-Part	AdaBoostM1-J48	AdaBoostM1-NBTree	Bagging	Bagging-Part	Bagging-J48	Bagging-NBTree	RacedIncrementalLogitBoost	RandomCommittee	RotationForest-Part	RotationForest-J48
2j9i:B				✓											
2jpc:A						✓									
2jqz:A									✓						
2jt1:A										✓					
2kkg:A			✓				✓								
2oug:A									✓						
2pkg:C										✓					
2pz9:A												✓			
2uvo:F			✓				✓								
2vpz:G				✓											
2w80:A			✓				✓								
2wyu:C												✓			
2zca:B									✓						
2ziy:A												✓			
2zkr:9					✓										
3by5:A	✓														
3eqs:A					✓										
3jyw:P				✓											
3kaw:F				✓											
3nk6:A												✓			
3o70:A								✓							

En la tabla se han sombreado de gris las columnas correspondientes a los modelos que mejor porcentaje de clasificación de sub-secuencias de aminoácidos negativos dieron, tal como se puede ver en la comparativa de modelos según la clasificación de sub-secuencias de aminoácidos negativos con el fichero de entrenamiento. Estos se considerarán como modelos *buenos*, mientras que el resto se considerarán como modelos *malos*.

Como 4 de los 9 modelos *buenos* (Bagging-NBTree, RandomCommittee, RotationForest-Part y RotationForest-J48) clasificaban muy pocas sub-secuencias como positivas hemos decidido no tenerlos en cuenta para estos resultados.

Las proteínas que destacan en varios modelos:

Tabla 53: tabla resumen con las proteínas negativas que destacan por el porcentaje de sub-secuencias de aminoácidos clasificados como positivos, en más de un modelo tras la aplicación de cada uno de los modelos generados sobre el fichero de proteínas negativas

Proteína	Cantidad de modelos *buenos* en los que destaca	Cantidad de modelos *malos* en los que destaca
1bcp:F	0	2
1dx1:A	2	0
1zza:A	0	2
2kkg:A	2	0
2uvo:F	2	0
2w80:A	2	0

Con esta clasificación ninguna proteína destaca en más de 2 modelos, por lo que parece más complicado poder sacar conjeturas más fiables sobre posibles proteínas erróneamente clasificadas.

5.5.3. Proteínas negativas con más probabilidad de ser positivas

En los apartados anteriores se han visto cuáles son las proteínas que más destacan según el criterio usado. Vamos a agrupar los resultados de ambos criterios usados para poder ver de un vistazo todas las proteínas inicialmente negativas que más probabilidad de ser positivas son, y así poder ver entre ellas cuáles son las que más destacan:

Tabla 54: proteínas que más han destacado unificando ambos criterios de selección

Proteína	Cantidad de modelos en los que destaca según criterio usado			
	Por total sub-secuencias		Por porcentaje de sub-secuencias	
	Modelos buenos	Modelos malos	Modelos buenos	Modelos malos
1bcp:F	0	0	0	2
1dx1:A	2	0	2	0
1zza:A	0	0	0	2
2aja:B	6	0	0	0
2gsx:A	0	3	0	0
2kkg:A	0	0	2	0
2q7z:A	0	3	0	0
2uvo:F	2	0	2	0
2vz8:B	1	6	0	0
2w80:A	0	0	2	0
2x2h:D	2	0	0	0
3cmu:A	2	3	0	0
3cmv:A	2	0	0	0
3cmw:A	2	2	0	0
3gau:A	2	1	0	0

Aunque estas 15 proteínas quizás merezcan la pena ser estudiadas en profundidad para poder determinar correctamente su estructura, parece que las que más probabilidad tienen de tener otra estructura debido a que destacan sobre el resto son:

- **1dx1:A**: por destacar en ambos criterios de selección en los modelos que mejor parecen clasificar las sub-secuencias, y que son más restrictivos.
- **2aja:B**: por destacar en 6 de los 9 modelos *buenos* en el criterio que parece más determinante.
- **2uvo:F**: por destacar en ambos criterios de selección en los modelos que mejor parecen clasificar las sub-secuencias, y que son más restrictivos.
- **2vz8:B**: por destacar en todos los modelos *malos*, y en uno de los *buenos*, en el criterio que parece más determinante.
- **3cmu:A**: por destacar en varios modelos *buenos* y *malos* en el criterio que parece más determinante.
- **3cmw:A**: por destacar en varios modelos *buenos* y *malos* en el criterio que parece más determinante.

5.6. Resumen de resultados de las proteínas inicialmente negativas que destacaban en el artículo previo

En la tabla de a continuación se han recopilado los datos del número de sub-secuencias que se han marcado como positivas en la aplicación de cada uno de los modelos sobre las proteínas que destacaban en el artículo previo functional and Genomic Analyses of Alpha-Solenoid Proteins[30]:

Tabla 55: resumen de las proteínas inicialmente negativas que destacaban en el artículo previo

	Part	J48	NBTree	AdaBoostM1	AdaBoostM1-Part	AdaBoostM1-J48	AdaBoostM1-NBTree	Bagging	Bagging-Part	Bagging-J48	Bagging-NBTree	RacedIncrementalLogitBoost	RandomCommittee	RotationForest-Part	RotationForest-J48
1iu1:B	1	1	0	4	0	0	0	2	0	0	0	8	0	0	0
1ozn:A	3	10	0	43	0	0	0	7	1	9	0	24	0	0	0
1r4x:A	0	8	0	9	0	0	0	7	0	9	0	19	0	0	0
1u6g:A	0	16	0	43	0	0	0	16	0	10	0	66	0	0	0
2dba:A	0	2	0	15	0	0	0	2	0	2	0	10	0	0	0
2e9g:A	1	3	0	9	0	0	0	3	0	3	0	7	0	0	0
2iv9:A	0	5	0	7	0	0	0	1	0	4	0	7	0	0	0
2vgl:M	2	13	0	17	0	0	0	2	0	6	0	38	0	0	0
2vgl:S	1	3	0	3	0	0	0	1	0	5	0	15	0	0	0
2z5k:A	4	32	0	70	2	1	0	15	1	18	0	84	0	0	0
2zej:B	1	5	0	10	0	0	0	3	0	3	0	14	0	0	0
3ifq:C	1	0	0	3	0	0	0	0	0	0	0	3	0	0	0
3lqv:P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Como se puede ver, en los modelos *buenos* (más restrictivos) ninguna de las proteínas, salvo la **2z5k:A** y en un caso la **1ozn:A**, tiene sub-secuencias marcadas como positivas. La proteína **2z5k:A** podría ser candidata a ser realmente positiva en este trabajo si los requisitos que se han ido usando en cada modelo se rebajasen un poco.

Si atendemos a los modelos *malos* (menos restrictivos) todas las proteínas, salvo la **3lqv:P**, tienen sub-secuencias positivas en varios modelos.

[30] Tabla S3 del artículo. Enlace directo: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0079894#pone.0079894.s003

6. Conclusiones

Aunque con este proceso que hemos seguido no podemos asegurar que ninguna de las proteínas inicialmente clasificadas como negativas es realmente positiva, hay varias proteínas que destacan al tener características similares a las que tienen las proteínas que se saben que son positivas, por lo que pueden merecer un estudio más particular de cada una de ellas para determinar la estructura real de cada una de ellas.

Atendiendo a los resultados obtenidos parece que el criterio de selección de proteínas basado en el número de total de sub-secuencias clasificadas como positivas proporciona mejores resultados, puesto que parece que clasifica las proteínas candidatas en más modelos, por lo que las proteínas que destacan no parecen ser fruto del azar sino por el hecho de compartir más características con las proteínas positivas del fichero de entrenamiento.

Las 5 posiciones de las sub-secuencias que parecen más importantes para determinar la clase de la sub-secuencia son, en este orden, la 25, 29, 13, 36 y 32. Es curioso que entre las 5 posiciones más importantes no se encuentre la posición 20 (justo mitad de la secuencia). Y a destacar también que las primeras posiciones no parecen muy relevantes en la clasificación de la sub-secuencia.

Por otro lado, el hecho de que una de las proteínas (2z5k:A) que destacaban en el artículo previo functional and Genomic Analyses of Alpha-Solenoid Proteins también destaque, aunque en menor medida que otras proteínas, en este trabajo nos hace pensar que la estrategia seguida puede ser buena para los fines buscados, y que las proteínas que más destacan en este trabajo podrían merecer la pena ser estudiadas.

A nivel personal este proyecto me ha permitido ampliar considerablemente mis conocimientos en el terreno del análisis de datos, además de haber puesto en práctica conocimientos de programación para poder realizar los pequeños programas que han sido necesarios para transformar y agrupar datos. Por otra parte, también me ha servido para adquirir algunos conocimientos sobre el mundo de las proteínas.

7. Trabajos futuros

En este proyecto se han usado sólo algunos algoritmos, y combinaciones de ellos, de entre todos los existentes actualmente en la herramienta Weka. En caso de que algún otro algoritmo destacase en la obtención de resultados positivos en otras situaciones similares se podría realizar el mismo proceso usándolo y comprobar si alguna de las proteínas que se han destacado en este trabajo destaca también en el nuevo algoritmo, o si destaca alguna otra proteína.

También podría realizarse este mismo trabajo pero partiendo de otro conjunto de proteínas en el fichero de entrenamiento que tenga una mayor nivelación entre las sub-secuencias positivas y las negativas, para no tener que repetir el conjunto de las positivas y que así tengan relevancia a la hora de generar los modelos.

Por supuesto, se puede repetir el proceso seguido con otros conjuntos de proteínas inicialmente negativas con la intención de detectar posibles proteínas candidatas erróneamente clasificadas.

En este trabajo nos hemos centrado en los datos asociados a la estructura de la proteína ya que ha sido el aspecto de las mismas que nos interesaban, pero también podría realizarse el mismo proceso usando datos de cualquier otro aspecto de las proteínas, u otros elementos biológicos, siempre que exista un conjunto amplio de datos base y una cierta incertidumbre sobre el aspecto en cuestión que haga interesante realizar el experimento.

8. Anexo

8.1.Bibliografía

[1] Functional and Genomic Analyses of Alpha-Solenoid Proteins
Fournier D, Palidwor GA, Shcherbinin S, Szengel A, Schaefer MH, et al. (2013) Functional and Genomic Analyses of Alpha-Solenoid Proteins. PLoS ONE 8(11): e79894. doi: 10.1371/journal.pone.0079894
http://dx.doi.org/10.1371%2Fjournal.pone.0079894

8.1.1. Algoritmos usados en la preparación de modelos

PART
[2] Eibe Frank and Ian H. Witten (1998). Generating Accurate Rule Sets Without Global Optimization. In Shavlik, J., ed., Machine Learning: Proceedings of the Fifteenth International Conference, Morgan Kaufmann Publishers, San Francisco, CA.
http://www.cs.waikato.ac.nz/~eibe/pubs/ML98-57.ps.gz

J48
[3] Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

NBTree
[4] Ron Kohavi: Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In: Second International Conference on Knowledge Discovery and Data Mining, 202-207, 1996.

AdaBoostM1
[5] Yoav Freund, Robert E. Schapire: Experiments with a new boosting algorithm. In: Thirteenth International Conference on Machine Learning, San Francisco, 148-156, 1996.
http://www.sciencedirect.com/science/article/pii/S002200009791504X

[6] Yoav Freund, Robert E. Schapire : Experiments with a New Boosting Algorithm
http://web.eecs.utk.edu/~leparker/Courses/CS425-528-fall10/Handouts/AdaBoost.M1.pdf

[7] Entrada en la Wikipedia: https://en.wikipedia.org/wiki/AdaBoost

Bagging
[8] Leo Breiman (1996). Bagging predictors. Machine Learning. 24(2):123-140.
http://link.springer.com/article/10.1007%2FBF00058655

RacedIncrementalLogitBoost

[9] Eibe Frank, Geoffrey Holmes, Richard Kirkby, Mark Hall: Racing committees for large datasets. In: Proceedings of the 5th International Conferenceon Discovery Science, 153-164, 2002.
http://www.cs.waikato.ac.nz/pubs/wp/2002/uow-cs-wp-2002-03.pdf
http://www.cs.waikato.ac.nz/~eibe/pubs/Frank_et_al_DS_2002.pdf

RandomCommittee
[10] Eibe Frank (eibe@cs.waikato.ac.nz)

[11] Documentación del modelo para la herramienta Weka.
http://weka.sourceforge.net/doc.stable/weka/classifiers/meta/RandomCommittee.html

RotationForest
[12] Rodriguez, J.J. ; Escuela Politecnica Superior, Burgos Univ. ; Kuncheva, L.I. ; Alonso, C.J.: Rotation Forest: A New Classifier Ensemble Method
http://www.computer.org/csdl/trans/tp/2006/10/01677518-abs.html
http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1677518

Comparaciones de clasificadores
[15] Reza Entezari-Maleki, Arash Rezaei, Behrouz Minaei-Bidgoli: Comparison of classification methods based on the type of attributes and sample size
http://www4.ncsu.edu/~arezaei2/paper/JCIT4-184028_Camera%20Ready.pdf

[16] Rich Caruana, Alexandru Niculescu-Mizil: An empirical comparison of supervised learning algorithms
http://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf

[17] Camilo Fitzgerald: A comparison of WEKA classification algorithms
https://camilofitzgerald.wordpress.com/2012/11/23/a-comparison-of-weka-classification-algorithms/

[13] Aprendizaje basado en árboles de decisión.
https://es.wikipedia.org/wiki/Aprendizaje_basado_en_%C3%A1rboles_de_decisi%C3%B3n

8.1.2. Evaluadores de atributos aplicados en los modelos generados en Weka

[29] ChiSquaredAttributeEval
Freund, J., Miller, I., & Miller, M. (2000) Estadística matemática con aplicaciones. Pearson –Prentice-Hall. Sexta edición.

[30] GainRatioAttributeEval
http://weka.sourceforge.net/doc.dev/weka/attributeSelection/GainRatioAttributeEval.html

[31] InfoGainAttributeEval
Lorenzo J. (2002). Selección de Atributos en Aprendizaje Automático basado en la Teoría de la Información. PhD thesis. Faculty of Computer Science, Univ. of Las Palmas. Gran Canaria.

[32] OneRAttributeEval
Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, *11*, 63-91

8.1.3. Proteínas, aminoácidos, etc

[18] Nomenclatura de aminoácidos
http://es.wikipedia.org/wiki/Nomenclatura_de_amino%C3%A1cidos

[21] Aminoácido
http://es.wikipedia.org/wiki/Amino%C3%A1cido

[22] Secuencia de aminoácidos
http://es.wikipedia.org/wiki/Secuencia_de_amino%C3%A1cidos

[23] Estructura de las proteínas
http://es.wikipedia.org/wiki/Estructura_de_las_prote%C3%ADnas

[24] Prof. Henry Jakubowski: Introduction to Amino Acids
http://biowiki.ucdavis.edu/Biochemistry/Proteins/Structure_and_Properties_of_Amino_Acids/Introduction_to_Amino_Acids

[25] Prof. Henry Jakubowski: Protein Conformation
http://biowiki.ucdavis.edu/Biochemistry/Proteins/Protein_Conformation

[26] Dr. Steve Carman: Amino Acids: An Introduction to Their Structure, Functions and Biochemical Properties
http://www.drcarman.info/bio223lb/223lab01.pdf

[27] The Columbia Electronic Encyclopedia: protein: Introduction
http://www.infoplease.com/encyclopedia/science/protein.html

[28] The Columbia Electronic Encyclopedia: protein: Protein Structure
http://www.infoplease.com/encyclopedia/science/protein-protein-structure.html

8.1.4. Herramienta Weka

[19] Hall, M. A., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The WEKA Data Mining Software: An Update. SIGKDD Explorations, 11 (1), 10-18.
Página principal de la herramienta: http://www.cs.waikato.ac.nz/ml/weka/

8.1.5. Otras fuentes

Trabajo previo no relacionado con este pero con uso de Weka para la búsqueda de resultados sobre grandes conjuntos de datos:

[14] Moreno, Valentín: Análisis de los criterios de relevancia documental mediante consultas de información en el entorno web
http://e-archivo.uc3m.es/bitstream/handle/10016/9727/2010_Valentin_Pelayo_-_tesis.pdf?sequence=1

[20] Información sobre el formato FASTA
http://es.wikipedia.org/wiki/Formato_FASTA

8.2.Presupuesto

8.2.1. Recursos

Para poder calcular correctamente el presupuesto del proyecto debemos identificar todos los recursos que intervienen en el mismo, ya sean humanos, de software o hardware

A continuación se detallan todos los recursos que han sido necesarios

8.2.1.1. Recursos hardware

Recurso	Coste (€)
Ordenador	800
Total	**800**

8.2.1.2. Recursos software

Recurso	Coste (€)
Eclipse IDE	0
Weka	0
Licencia Microsoft Office 2010	200
Licencia Windows 7	250
Total	**450**

8.2.1.3. Recursos humanos

Para el cálculo de los recursos humanos se ha considerado:

- 1 analista programador
- Coste bruto por cada hora de un analista programador: 25€

Recurso	Horas	Coste (€)
Fase de Análisis		
Recopilación de información previa	30	25
Estudio del problema	100	25
Análisis de la información y formatos	35	25
Análisis formatos de entrada	25	25
Total fase de análisis:	190	4.750 €
Fase de diseño e implementación		
Creación de programas necesarios para la adaptación y transformación de datos de entrada	80	20
Creación de programas necesarios para la agrupación parcial de resultados	60	20

Total fase de implementación:	140	2.800 €
Fase de documentación		
Documentación realizada	220	20
Total fase de documentación:	220	4.400 €
Total	**550**	**11.950 €**

8.2.2. Resumen

Como resumen de lo detallado en el punto anterior tenemos:

Tipo de recurso	Coste (€)
Hardware	800
Software	450
Humanos	11.950
Total	**13.200 €**

8.3.Valores válidos para un aminoácido en el formato FASTA

Fuente [20]

Valor	Significado
A	Alanina
B	Ácido aspártico o Asparagina
C	Cisteína
D	Ácido aspártico
E	Ácido glutámico
F	Fenilalanina
G	Glicina
H	Histidina
I	Isoleucina
K	Lisina
L	Leucina
M	Metionina
N	Asparagina
O	Pirrolisina
P	Prolina
Q	Glutamina
R	Arginina
S	Serina
T	Treonina
U	Selenocisteína
V	Valina
W	Triptófano
Y	Tirosina
Z	Ácido glutámico o Glutamina
X	cualquiera
*	parada de traducción
-	hueco (gap) de longitud indeterminada

8.4.Uso de la herramienta Weka

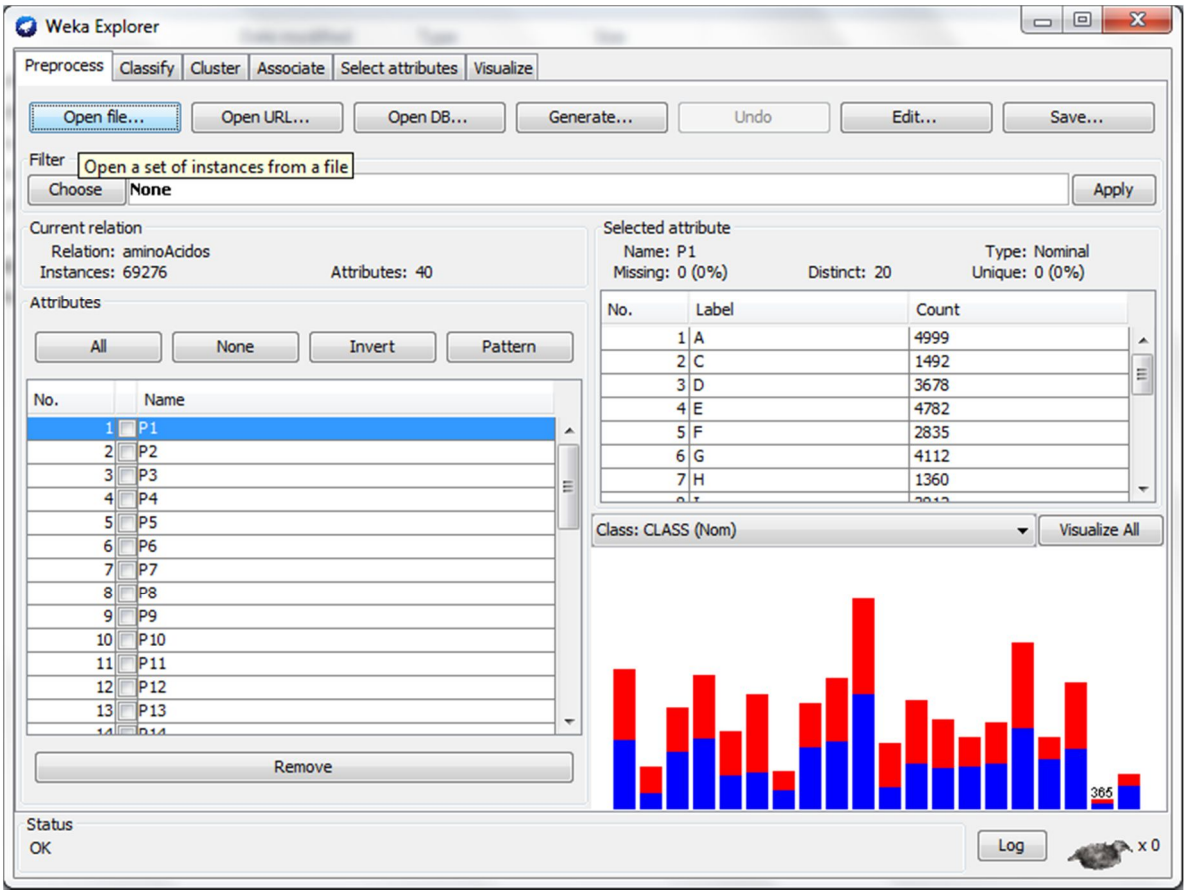
En este apartado vamos a explicar cómo usar la herramienta Weka, indicando paso a paso cada una de las funcionalidades que hemos usado para este proyecto.

8.4.1. Cómo generar un modelo en Weka a partir de un fichero de entrenamiento

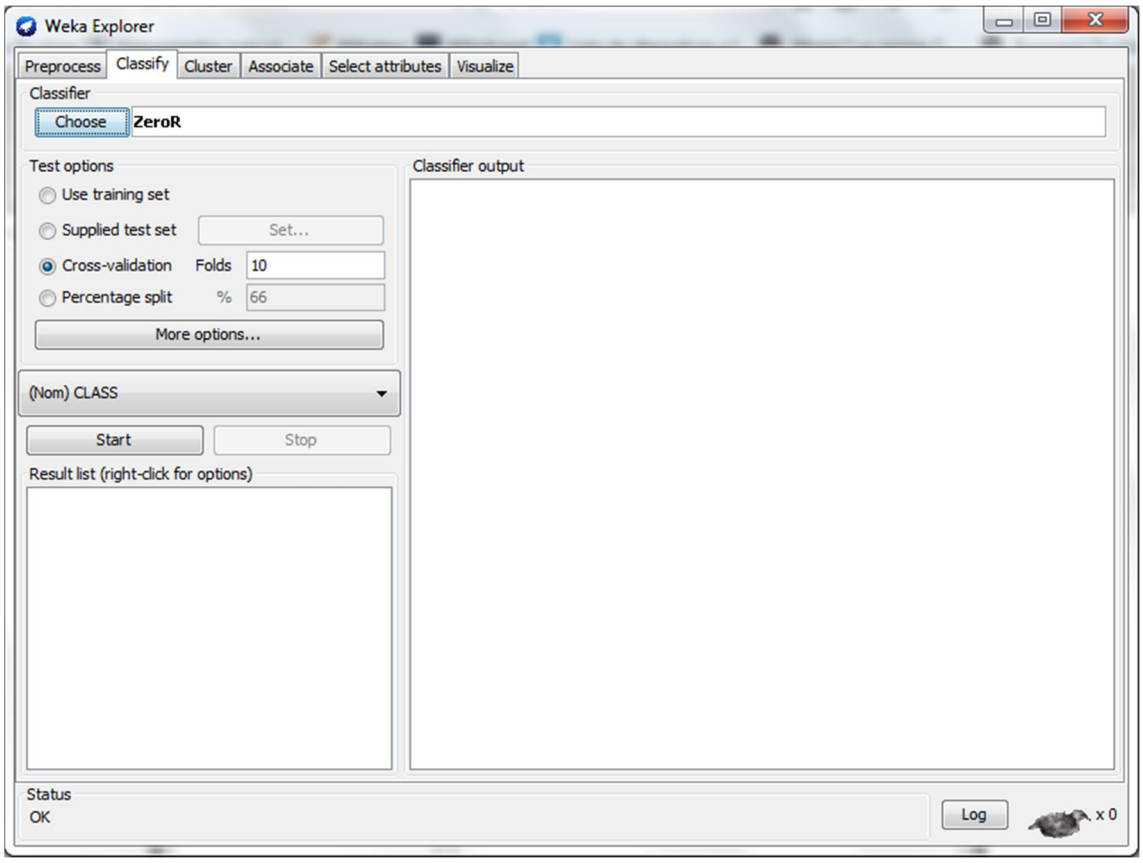
Los pasos a seguir en la herramienta Weka para generar un modelo a partir de un fichero de entrenamiento

Abrimos el programa y le damos a la opción "Explorer":

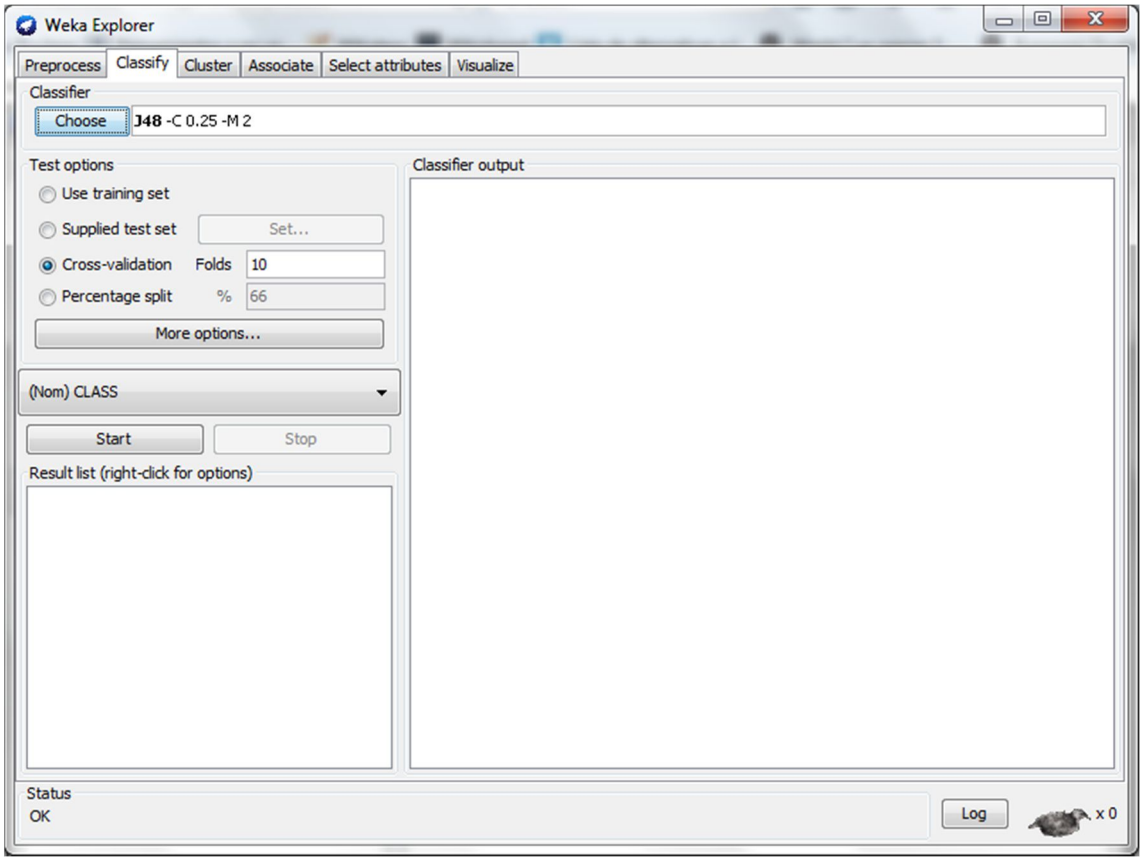
En la ventana que aparece le damos a "Open file…" y seleccionamos el fichero de entrenamiento en formado *.arff:



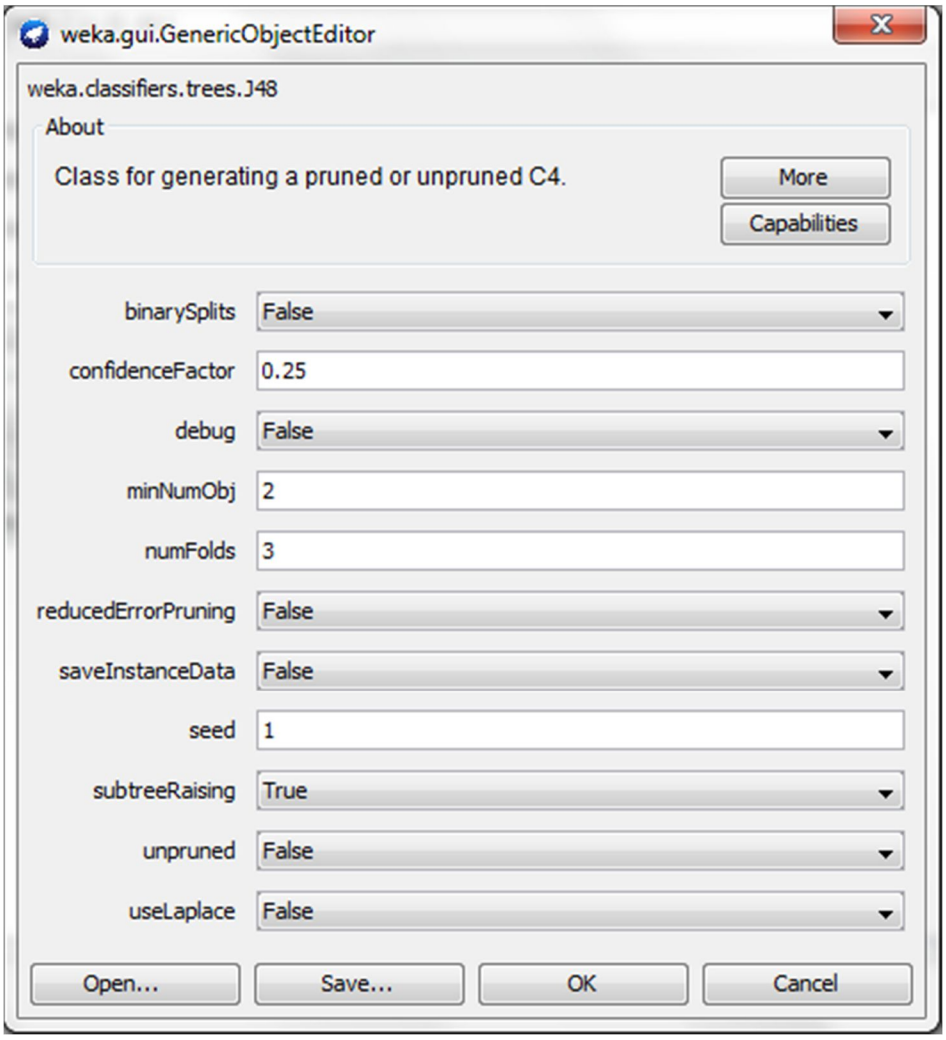
Ahora vamos a la pestaña "Classify":



Le damos al botón "Choose" situado en la parte de arriba, dentro de la sección "Classifier" y seleccionamos el clasificador (algoritmo) que deseemos:



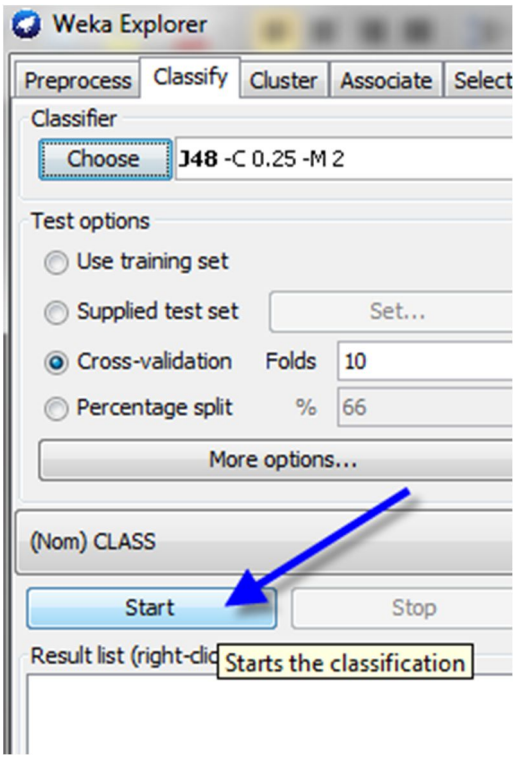
Cuando se selecciona se vuelve a la ventana anterior con el clasificador actualizado. Si se desease cambiar los valores por defecto del clasificador basta con clickar con el ratón en el filtro y se abre una ventana nueva con los valores actuales que pueden ser cambiados:



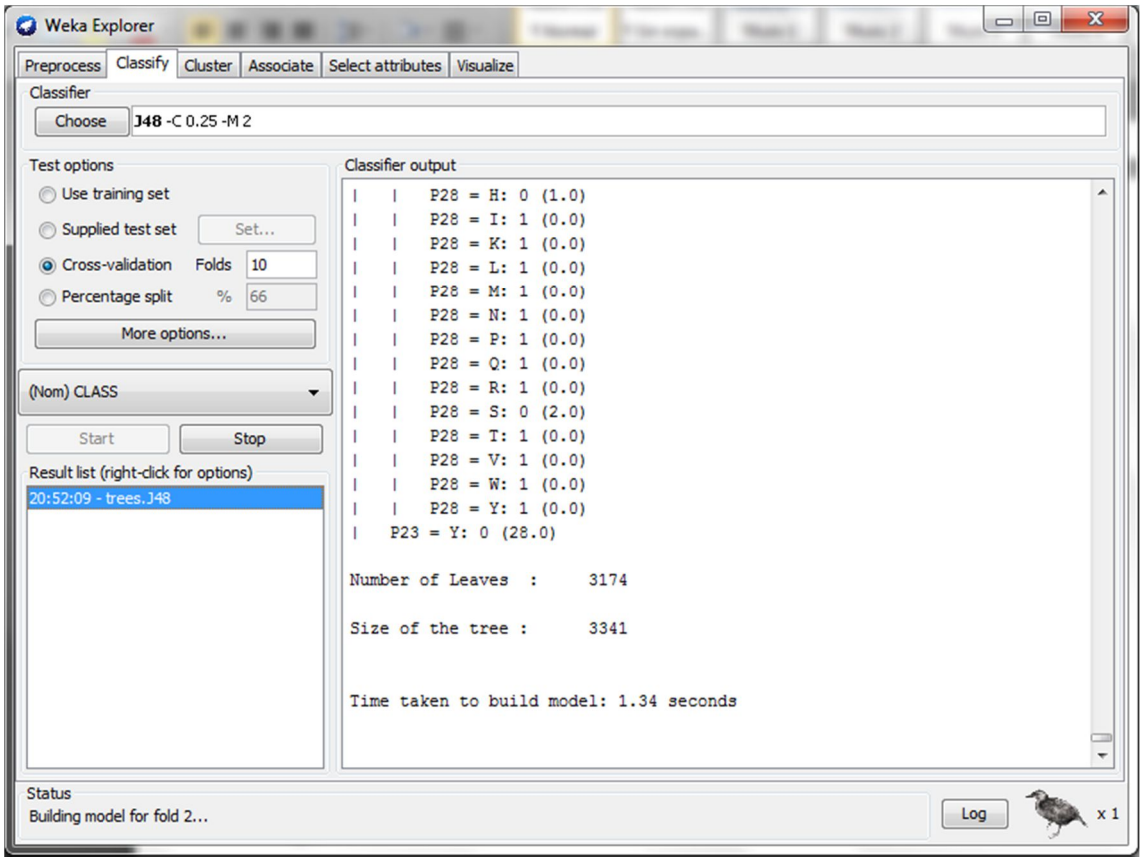
Una vez cambiados los valores oportunos se vuelve a la ventana anterior.

Nota: para este proyecto no se han cambiado los valores por defecto de ninguno de los algoritmos de clasificación usados ya que hemos considerado que dichos valores son los más adecuados para nuestros modelos.

Ahora basta con dar al botón "Start":

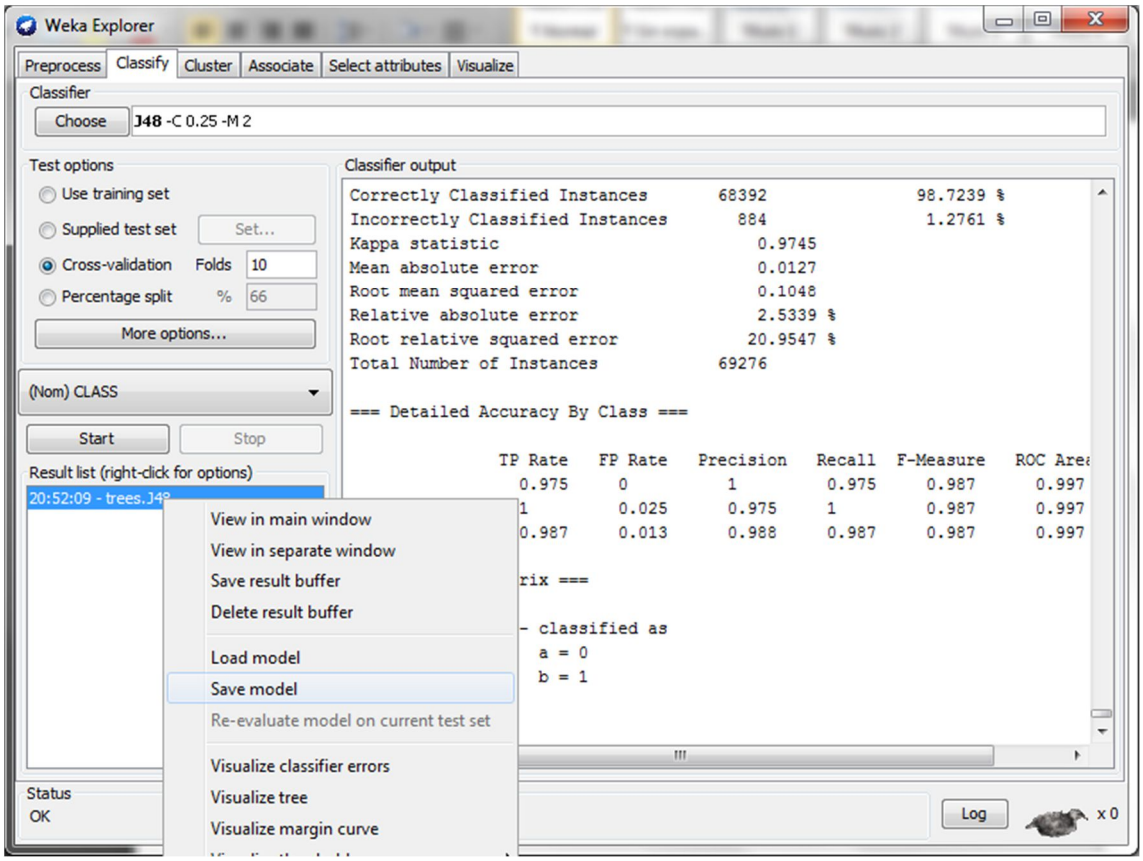


Y comienza la generación del modelo:



Cuando termina el proceso podemos ver en la parte principal ("Classifier output") información sobre el modelo generado, incluyendo las reglas que forman el modelo y unos datos resumen sobre el mismo

Además, se puede exportar el modelo para poder usarlo posteriormente –como se ha hecho- dándole al botón derecho sobre el nombre del modelo de la parte baja-izquierda y pulsando sobre la opción "Save model":

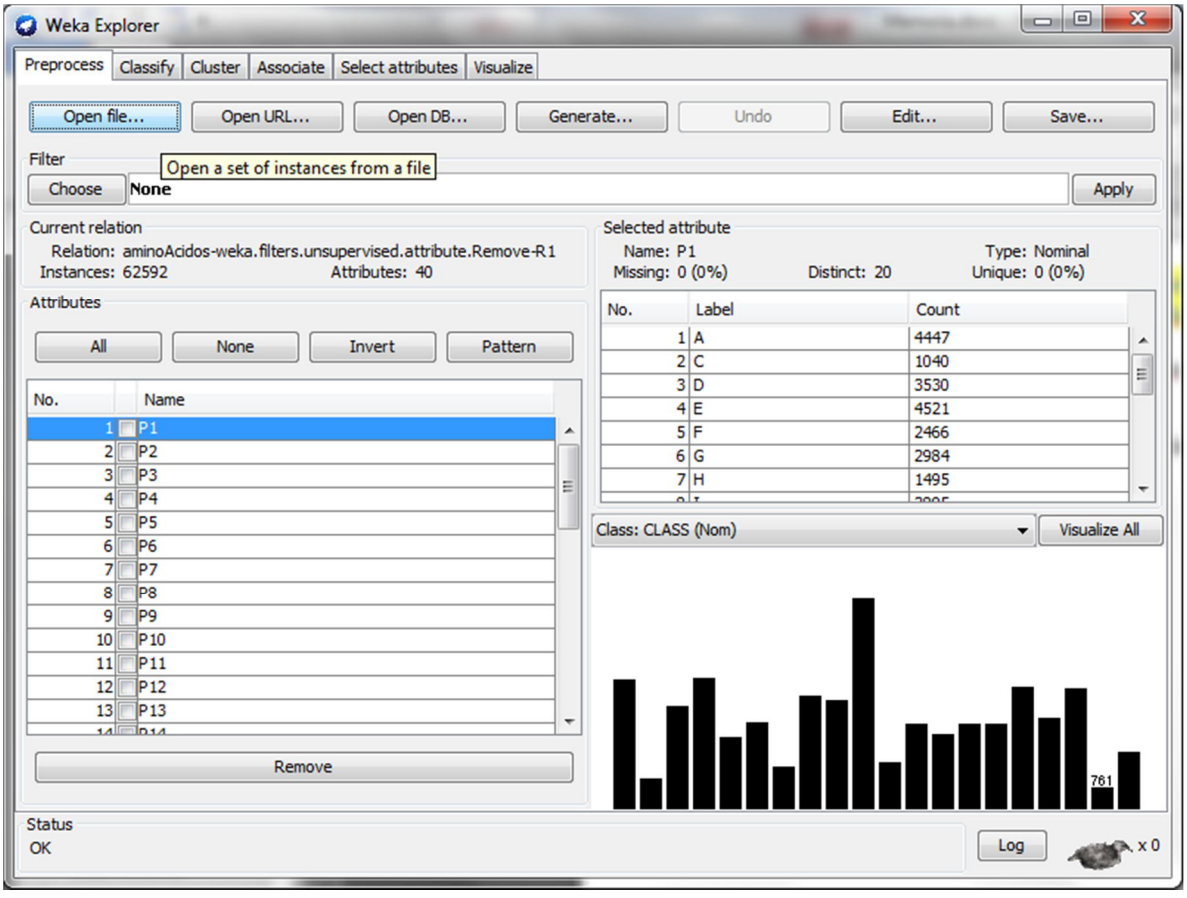


8.4.2. Cómo aplicar un modelo generado anteriormente sobre un fichero de entrada en Weka

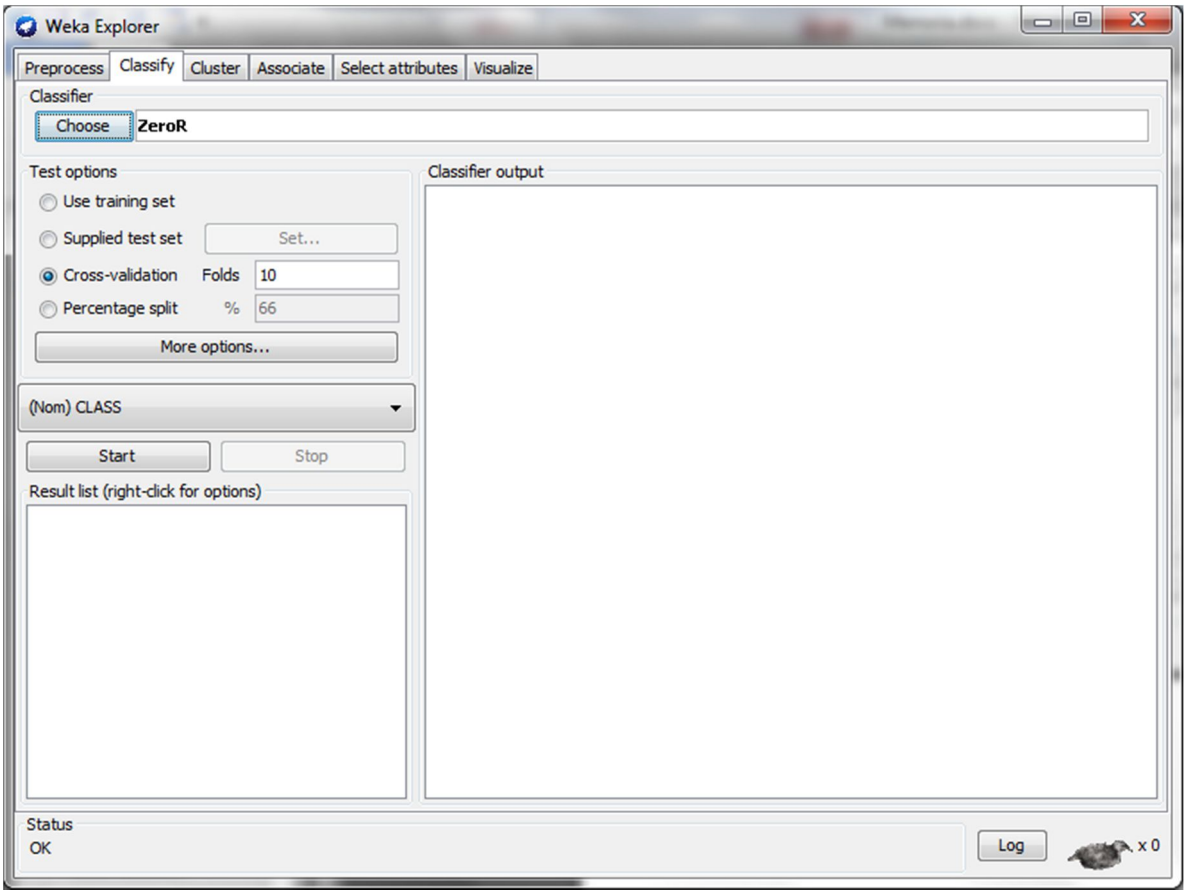
Los pasos a seguir en la herramienta Weka para aplicar un modelo generado previamente sobre un fichero en formato Weka:

Abrimos el programa y le damos a la opción "Explorer":

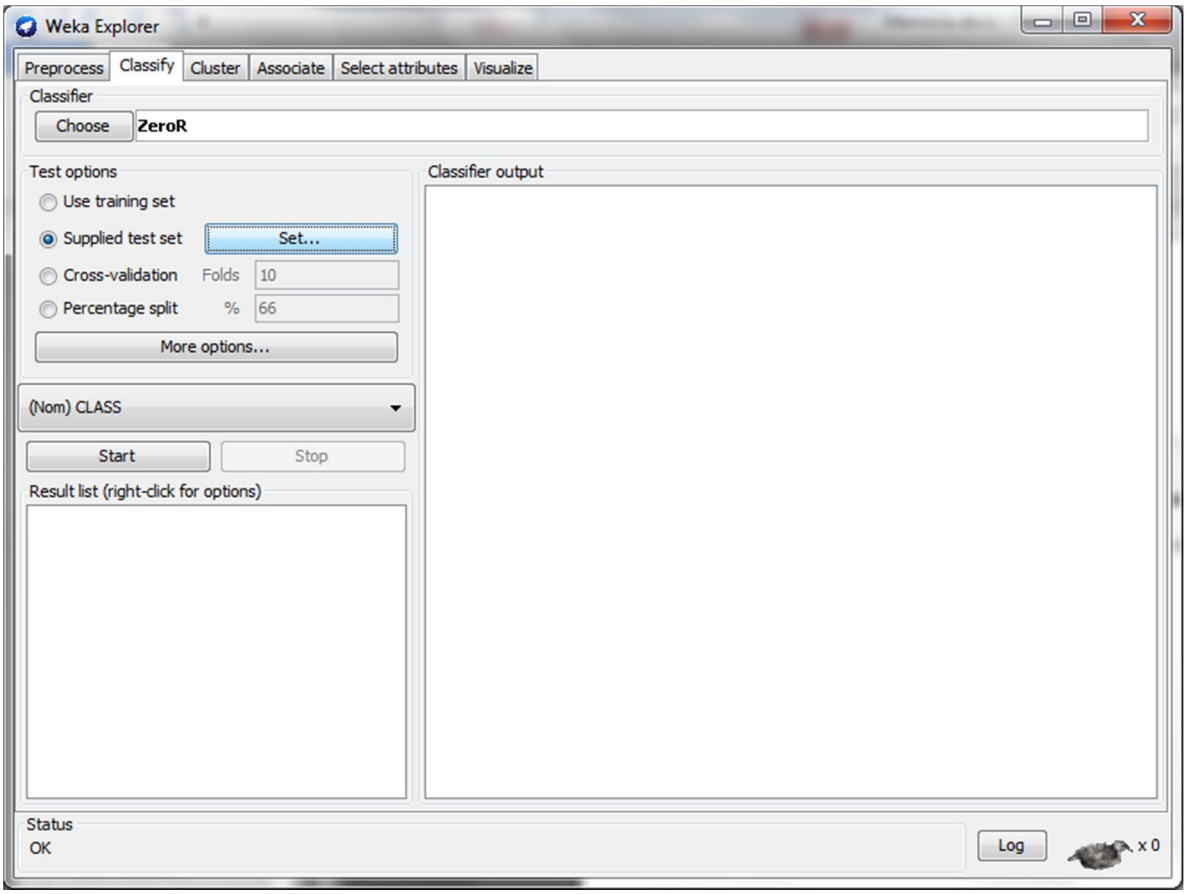
En la ventana que aparece le damos a "Open file…" y seleccionamos el fichero de entrada en formado *.arff:



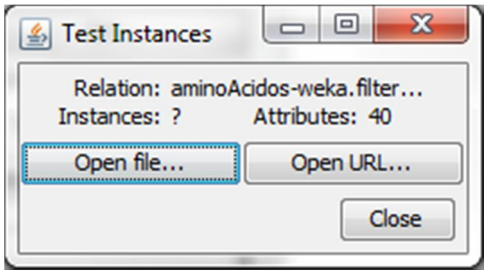
Vamos a la pestaña "Classify":



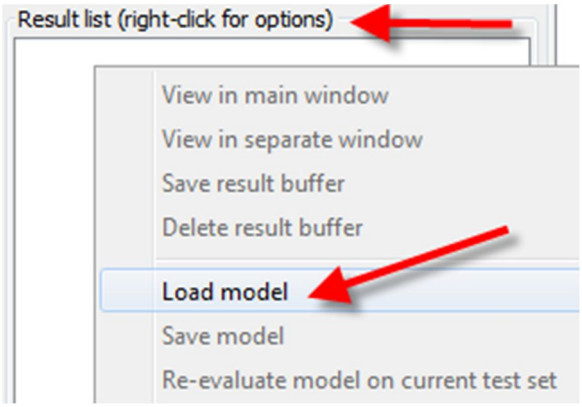
Dentro de la sección "Test options" hay que seleccionar "Supplied test set" y dar al botón "Set…" que ahora estará activo:



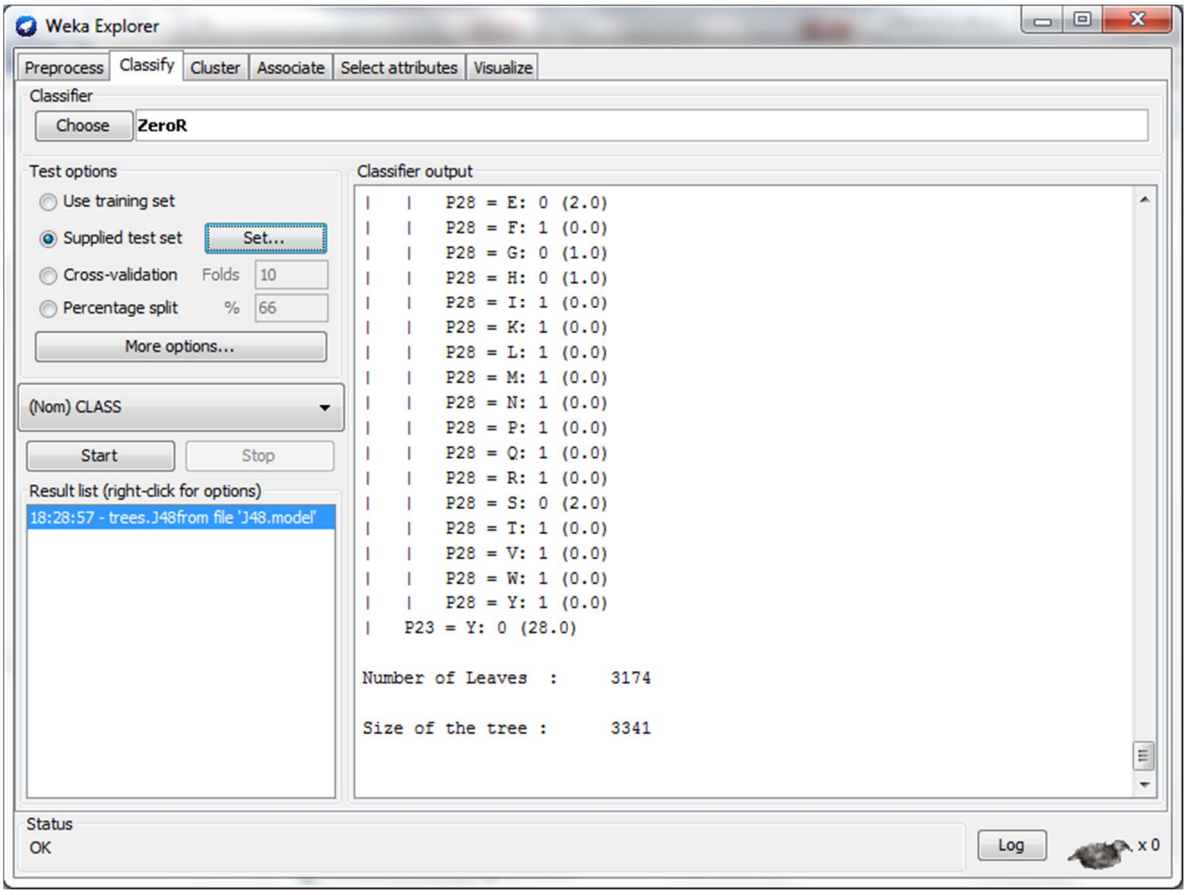
En la pequeña ventana que se abre le damos a "Open file" y buscamos de nuevo el fichero de entrada que deseemos tratar con el modelo, y una vez hecho le damos a "Close":



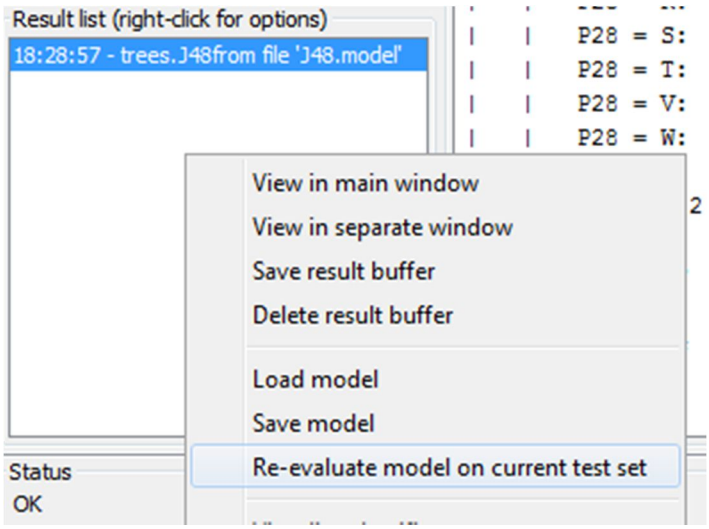
Ahora dentro del área "Result list" hay que dar al botón derecho del ratón para ver más opciones, y darle a "Load model":



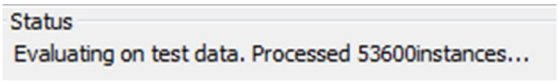
Buscamos el fichero *.model creado anteriormente y lo abrimos. Una vez que Weka lo ha cargado nos quedará algo como esta ventana:



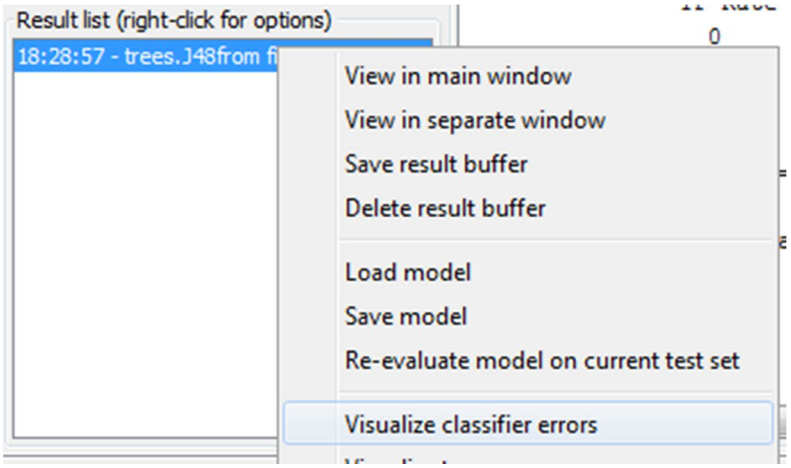
Ahora hay que ir de nuevo al área "Result list" y dar al botón derecho del ratón, esta vez para seleccionar la opción "Re-evaluate modelo n current test set":



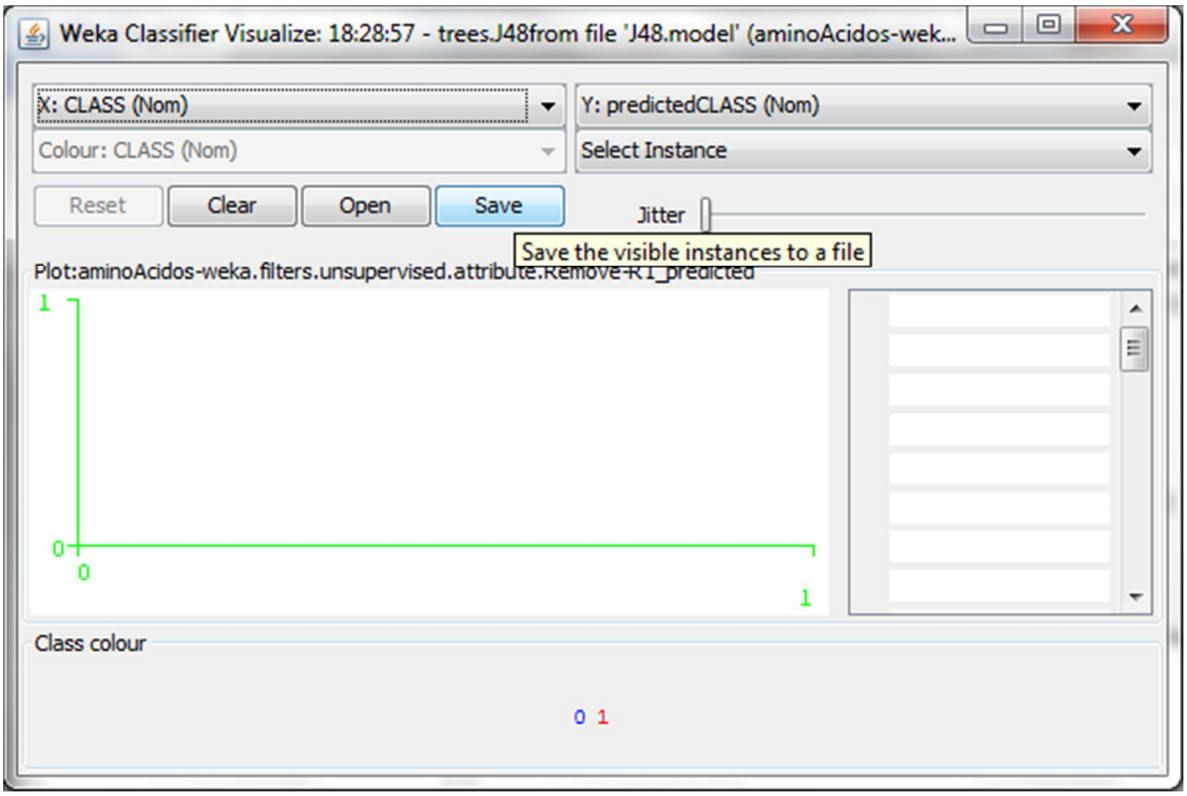
Se pondrá en marcha el proceso. Podemos ver el progreso en la barra inferior:



Para exportar el fichero *.arff incluyendo los resultados calculados por cada entrada hay que volver al área "Result list", dar al botón derecho y seleccionar la opción "Visualize classifier errors":



En la nueva ventana hay que darle al botón "Save":



8.5. Significado de los campos que se usan en la generación de un modelo en Weka

Los campos son:

- *binarySplits*: indica si debe hacer divisiones binaria en atributos nominales mientras construye el árbol parcial.
- *confidenceFactor*: el factor de confianza usado para la poda del árbol (valores pequeños provoca más poda).
- *debug*: indica si debe sacar más información o no durante el proceso.
- *minNumObj*: número mínimo de instancias por regla.
- *numFolds*: determina la cantidad de datos usados para la poda de error-reducido.
- *reducedErrorPruning*: indica si usa la poda de error-reducido en lugar de la poda del árbol temporal.
- *seed*: semilla usada para hacer aletorio los datos que se usan en la poda de error-reducido.
- *unpruned*: indica si debe realizar la poda.